



**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ  
ΓΙΑ ΤΗ ΔΙΕΡΕΥΝΗΣΗ ΔΥΝΑΤΟΤΗΤΩΝ ΠΡΟΣΒΑΣΙΜΟΤΗΤΑΣ  
ΣΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ WINDOWS MOBILE**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
ΤΟΥ  
ΚΩΝΣΤΑΝΤΙΝΟΥ Π. ΤΟΛΙΑ**

**Επιβλέπων:**

Τσεκούρας Γιώργος  
Επίκουρος Καθηγητής Πανεπιστημίου Αιγαίου

**Συνεπιβλέποντες:**

1. Μπαντιμαρούδης Φιλήμων  
Μόνιμος Επίκουρος Καθηγητής Πανεπιστημίου Αιγαίου
2. Καμμάς Σταύρος  
Διδάσκων ΠΔ 407/80 Πανεπιστημίου Αιγαίου

*Μυτιλήνη, Σεπτέμβριος 2010*

## Εισαγωγή

Η παρούσα εργασία, αφορά στο σχεδιασμό και στην ανάπτυξη πολυμεσικής πολιτιστικής εφαρμογής για το λειτουργικό σύστημα της εταιρείας λογισμικού Microsoft για κινητές συσκευές (Mobile Devices), Windows Mobile. Χωρίζεται στις εξής ενότητες:

- Στην ανάλυση βασικών εννοιών, ορισμών και δεδομένων που σχετίζονται με τις αναπηρίες, τις αρχές της καλής σχεδίασης συστημάτων και προϊόντων λογισμικού και την προσβασιμότητα σε αυτές.
- Στην κατανόηση γύρω από τα εργαλεία υλοποίησης και ανάπτυξης της εφαρμογής.
- Στην ανάπτυξη και την παρουσίαση της εφαρμογής.
- Στα τελικά αξιολόγηση και τα συμπεράσματα που εξάγονται από την ερευνητική εργασία.
- Τέλος, παρατίθεται το παράρτημα με τον κώδικα που συγγράφηκε για τη λειτουργία της εφαρμογής.

Πιο αναλυτικά:

Το πρώτο μέρος, αναφέρεται στο τι εννοούνται σήμερα για τον τομέα της Τεχνολογίας Πληροφοριών και Επικοινωνιών (ΤΠΕ) οι έννοιες γύρω από τις αναπηρίες και τις αρχές καλής σχεδίασης. Δίνονται γενικά στοιχεία γύρω από τη νομοθεσία που υπάρχει σήμερα για την πρόσβαση των ΑμεΑ στις ΤΠΕ, όπως και γιατί τελικά η υιοθέτηση κανόνων προσβασιμότητας κρίνεται απαραίτητη.

Το δεύτερο μέρος ενημερώνει τον αναγνώστη γύρω από όλα τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής, είτε αφορούν λογισμικό ή συσκευές, είτε μεθόδους και γλώσσες προγραμματισμού. Ποια είναι η χρησιμότητά τους, οι δυνατότητές τους, ποιο ρόλο αυτά επιτελούν. Επίσης με ποια εργαλεία είναι δυνατόν η εφαρμογή που αναπτύχθηκε να παρουσιαστεί για να αξιοποιηθούν οι δυνατότητές της.

Μετά αναλύεται η υλοποίηση της εφαρμογής. Ποιο μοντέλο χρησιμοποιήθηκε και πώς αυτό κρίθηκε ως το καταλληλότερο, ποιες απαιτήσεις λήφθηκαν υπ' όψιν. Έπειτα, στην παρουσίαση της εφαρμογής, παρατίθενται οι μέθοδοι πλοήγησης, οι φόρμες από τις οποίες αυτή αποτελείται. Γίνεται μια εξοικείωση του χρήστη με το περιβάλλον της κι η παρουσίαση των κανόνων προσβασιμότητας που ενσωματώθηκαν, σύμφωνα και με τα όσα αναφέρθηκαν στο 1<sup>ο</sup> κεφάλαιο. Γίνεται ενημέρωση γύρω από το σύνολο των δυνατοτήτων της εφαρμογής, ενώ αναγράφονται και παραδείγματα κώδικα για το πώς όλες οι παραπάνω διαδικασίες πραγματοποιούνται.

Τέλος, το κείμενο έρχεται να κλείσει, πέρα από την παράθεση του κώδικα, η αξιολόγηση γύρω από το ερώτημα του κατά πόσο επετεύχθησαν οι στόχοι κι οι απαιτήσεις που τέθηκαν στο προηγούμενο κεφάλαιο. Τα τελικά επίσης συμπεράσματα κι οι συμβουλές προς τους αναγνώστες για την ανάπτυξη προσβάσιμων εφαρμογών για Mobile Devices.

# Πίνακας Περιεχομένων

<b>Κεφάλαιο 1: Εισαγωγή στις βασικές έννοιες</b>	<b>5</b>
<b>1.1</b> Σχετικά με τις αναπηρίες	<b>5</b>
<b>1.2</b> Κατηγοριοποίηση ικανοτήτων	<b>6</b>
<b>1.3</b> Βαθμός ανικανότητας	<b>6</b>
<b>1.4</b> Προσβασιμότητα	<b>7</b>
<b>1.5</b> Σχεδίαση για όλους	<b>7</b>
1.5.1 Καταρρίπτοντας τους μύθους της «σχεδίασης για όλους»	8
1.5.2 Η σημασία της σχεδίασης για όλους	9
1.5.3 Αρχές καθολικής σχεδίασης	9
1.5.4 Η καθολική πρόσβαση ή πρόσβαση για όλους	10
<b>1.6</b> Νομοθεσία	<b>10</b>
<b>1.7</b> Τελικά, γιατί προσβασιμότητα;	<b>11</b>
<b>Κεφάλαιο 2: Εργαλεία υλοποίησης και παρουσίασης εφαρμογής</b>	<b>12</b>
<b>2.1</b> Αντικειμενοστραφής προγραμματισμός	<b>12</b>
<b>2.2</b> Η γλώσσα προγραμματισμού Microsoft Visual C#	<b>13</b>
2.2.1 Η έκδοση Microsoft Visual C# 3.0	13
<b>2.3</b> Το πλαίσιο ανάπτυξης εφαρμογών Microsoft .NET Framework	<b>14</b>
2.3.1 Τα πλεονεκτήματα του πλαισίου ανάπτυξης εφαρμογών Microsoft .NET	16
2.3.2 Η έκδοση .NET Framework 3.5	17
<b>2.4</b> Η σχέση μεταξύ Microsoft .NET και Microsoft Visual C#	<b>18</b>
<b>2.5</b> Λογισμικό ανάπτυξης εφαρμογών Microsoft Visual Studio 2008	<b>18</b>
<b>2.6</b> Το λειτουργικό σύστημα Windows Mobile	<b>20</b>
2.6.1 Η έκδοση Windows Mobile 6.5	22
<b>2.7</b> Mobile Device – Smartphone	<b>23</b>
<b>2.8</b> Ανάπτυξη λογισμικού για το Windows Mobile	<b>25</b>
<b>2.9</b> Ανάπτυξη εφαρμογών πολιτιστικού πολυμεσικού περιεχομένου με δυνατότητες προσβασιμότητας για Windows Mobile	<b>26</b>

<b>Κεφάλαιο 3: Ανάπτυξη και Παρουσίαση της Εφαρμογής</b>	<b>27</b>
<b>3.1 Μοντέλο διαδικασίας λογισμικού που χρησιμοποιήθηκε</b>	<b>27</b>
3.1.1 Γιατί επαναληπτικό μοντέλο για την εφαρμογή;	28
<b>3.2 Επιλογή θέματος παρουσίασης για την εφαρμογή</b>	<b>29</b>
<b>3.3 Ανάλυση απαιτήσεων εφαρμογής</b>	<b>30</b>
3.3.1 Φυσικό περιβάλλον	30
3.3.2 Χρήστες κι εργονομικοί παράγοντες	30
3.3.3 Λειτουργικότητα	31
3.3.4 Λοιπές απαιτήσεις	31
<b>3.4 Πλοήγηση</b>	<b>31</b>
3.4.1 Βασική πλοήγηση	32
3.4.2 Εναλλακτική πλοήγηση	33
<b>3.5 Φόρμες εφαρμογής</b>	<b>34</b>
<b>3.6 Εξοικείωση του χρήστη με το περιβάλλον της εφαρμογής</b>	<b>35</b>
<b>3.7 Ενσωμάτωση αρχών προσβασιμότητας</b>	<b>35</b>
<b>3.8 Τεχνικές και μέθοδοι για την ικανοποίηση απαιτήσεων προσβασιμότητας</b>	<b>36</b>
<b>3.9 Αξιοποίηση των δυνατοτήτων της εφαρμογής</b>	<b>42</b>
<b>Κεφάλαιο 4: Αξιολόγηση Εφαρμογής</b>	<b>46</b>
<b>4.1 Αποτελέσματα- Αξιολόγηση</b>	<b>46</b>
<b>4.2 Συμπεράσματα</b>	<b>47</b>
<b>Παράρτημα</b>	<b>48</b>
<b>Βιβλιογραφία</b>	<b>109</b>

# Κεφάλαιο 1: Εισαγωγή στις βασικές έννοιες

---

## 1.1 Σχετικά με τις αναπηρίες

Με τον όρο αναπηρία εννοούμε το αποτέλεσμα οργανικών ή περιβαλλοντολογικών αιτιών, που δημιουργούν ένα σύνολο εμποδίων σε σημαντικές περιοχές της ζωής, όπως η αυτοεξυπηρέτηση, η απασχόληση, η εκπαίδευση, η ψυχαγωγία και η γενικότερη κοινωνική συμμετοχή.

Σήμερα, αναπηρία δεν θεωρείται κάποια κατάσταση ενός ανθρώπου, αλλά το αποτέλεσμα της σχέσης των ικανοτήτων ενός ατόμου και των απαιτήσεων του περιβάλλοντος.

Η μείωση της ικανότητας ενός ατόμου να πραγματοποιήσει μια λειτουργία ή δραστηριότητα ονομάζεται **κατάπτωση** ή **απώλεια ικανότητας**. Προβλήματα υγείας, εκ γενετής προβλήματα υγείας, η διαδικασία της γήρανσης, τραυματικά γεγονότα, μπορούν να οδηγήσουν σε ελάττωση των ικανοτήτων. Ανεξάρτητα από τα παραπάνω, απώλεια ικανοτήτων μπορεί να συμβεί σε ένα άτομο περιπτωσιακά ή παροδικά. Είτε αυτή η ελάττωση (μόνιμη, περιπτωσιακή ή παροδική) οδηγεί σε αναπηρία είτε όχι, καθορίζεται σίγουρα από κοινωνικούς και περιβαλλοντικούς παράγοντες, και το πιο σημαντικό από τον σχεδιασμό περιβαλλόντων, προϊόντων, συστημάτων και υπηρεσιών.

Η διεθνής βάση των αναπηριών δημοσιεύτηκε αρχικά από τον Παγκόσμιο Οργανισμό Υγείας το 1980. Στηρίχθηκε σε ένα διευρυμένο ιατρικό μοντέλο, το οποίο περιγράφει ένα σύνολο από αιτίες, ξεκινώντας από την κατάπτωση δεξιοτήτων (impairment) (απώλεια σταθερότητας ή λειτουργίας), που οδηγούν στην ανικανότητα (disability) (απώλεια των ικανοτήτων) και τελικά στην αναπηρία (handicap) (απώλεια της ικανότητας συμμετοχής σε κοινωνικές διεργασίες). Έτσι η έμφαση πέρασε από την ιατρική κατάσταση στην πρόοδο της αναπηρίας και στις περιβαλλοντικές και κοινωνικές συνθήκες υπό το πρίσμα των οποίων αυτή πραγματοποιείται.

Τον Νοέμβριο του 2001, μετά από έρευνα επτά χρόνων, το πρωτότυπο μοντέλο αντικαταστάθηκε από μία Διεθνή Κατηγοριοποίηση των Ικανοτήτων, της Αναπηρίας και της Υγείας (International Classification of Functioning, Disability and Health, ICF). Αυτή η αλλαγή εστιάζει την προσοχή της από την κατάσταση υγείας ενός ατόμου έως το πώς ζει με τις συγκεκριμένες «συνθήκες υγείας», με ποιο τρόπο αυτές μπορούν να βελτιωθούν ή να καλυτερέψουν, ώστε να αποκτήσει μια παραγωγική, γεμάτη ζωή. Με άλλα λόγια η προσοχή εστιάζεται περισσότερο στο πώς λειτουργούμε μέσα σε κοινωνικά ή άλλου είδους όρια, παρά στην κατάσταση υγείας του εκάστοτε ατόμου. Ακόμα πιο σημαντικό είναι ότι η προσοχή εστιάζεται στην αλληλεπίδραση ανάμεσα στην ιατρική κατάσταση,

τους περιβαλλοντικούς παράγοντες και τους προσωπικούς παράγοντες, αλλά και στον τρόπο με τον οποίο όλα τα παραπάνω επηρεάζουν τις σωματικές λειτουργίες και διεργασίες, και τελικά τις δραστηριότητες και τη συμμετοχή του ατόμου σε αυτές. <sup>[22]</sup>

## 1.2 Κατηγοριοποίηση ικανοτήτων

Οι ικανότητες που είναι αναγκαίες για την συμμετοχή κάποιου στις δραστηριότητες της καθημερινής ζωής είναι 13. Εφτά από αυτές τις ικανότητες είναι σημαντικές για τον σχεδιασμό προϊόντων και υπηρεσιών:

- όραση,
- ακοή,
- επικοινωνία,
- νοητικές λειτουργίες,
- μετακίνηση,
- επιδεξιότητα,
- τέντωμα/ φτάσιμο. <sup>[22]</sup>

## 1.3 Βαθμός ανικανότητας

Η συνολική ανικανότητα ενός ατόμου καθορίζεται από το σταθμισμένο άθροισμα των τριών πιο σοβαρών ανικανότητων του. Το απλό αυτό μοντέλο έχει την εξής μορφή:

- σταθμισμένη πλήρης ανικανότητα = η χειρότερη επίδοση + 0.4 X (δεύτερη χειρότερη επίδοση) + 0.3 X (τρίτη χειρότερη επίδοση)

### Βαθμός Ανικανότητας και Κατηγορίες Επίδοσης

Απώλεια Ικανοτήτων	Βαθμός Ανικανότητας	Βαθμός Επίδοσης	Βαθμός Ικανότητας
	0	0	Πλήρης ικανότητα
Χαμηλή	0.5 - 2.95	1	Αρκετά ικανοποιητική
	3 - 4.95	2	
	5 - 6.95	3	
Μέτρια	7 - 8.95	4	Μέτρια
	9 - 10.95	5	
	11 - 12.95	6	
Υψηλή	13 - 14.95	7	Χαμηλή
	15 - 16.95	8	
	17 - 18.95	9	
	19 - 21.4	10	

**Σχήμα 1-1:** Πίνακας Βαθμού Ανικανότητας και Κατηγοριών Επιδόσεων <sup>[22]</sup>

Οι βαθμοί επίδοσης σε κάθε μία από τις επτά κατηγορίες ικανοτήτων: όραση, ακοή, επικοινωνία, νοητικές λειτουργίες, μετακίνηση, επιδεξιότητα και τέντωμα / φτάσιμο μπορούν να υπολογιστούν με απλό τρόπο.

Το παραπάνω σταθμισμένο άθροισμα λαμβάνει τιμές από 0 ως 21,4 και σύμφωνα με τον παρακάτω πίνακα, αντιστοιχίζεται μετά σε μια κλίμακα από 0 ως 10 με καθορισμένα

στάδια, όπου το 0 αναπαριστά την πλήρη ικανότητα (απουσία αναπηρίας) και το 10 την ελάχιστη ικανότητα (σοβαρή αναπηρία). Οι επιδόσεις μπορούν να συνδυαστούν, ώστε να προκύψει μια κλίμακα τεσσάρων σταδίων, που να περιλαμβάνει: την πλήρη ικανότητα (0), την αρκετά ικανοποιητική (1-2) και τη μέτρια (3-6) ικανότητα, και την χαμηλή (7-10) ικανότητα.

Τα αποτελέσματα της εφαρμογής του παραπάνω μοντέλου στη Μεγάλη Βρετανία έδειξαν ότι περίπου 8.582.200 ενήλικες - το 20% του πληθυσμού των ενηλίκων - παρουσιάζει κάποιο είδος αναπηρίας σύμφωνα με τον ορισμό που χρησιμοποιείται. Το 34% των ατόμων αυτών έχει μικρό βαθμό κατάπτωσης των λειτουργιών (υψηλό επίπεδο ικανοτήτων), το 45% έχει μέτριο βαθμό κατάπτωσης των λειτουργιών (μέτριο επίπεδο ικανοτήτων) και το 21% έχει μεγάλο βαθμό κατάπτωσης των λειτουργιών (χαμηλό επίπεδο ικανοτήτων).<sup>[22]</sup>

Σύμφωνα επίσης με στοιχεία της εταιρείας Microsoft Corporation για τις αναπηρίες συνολικά στον πληθυσμό:

- 1 στα 4 άτομα έχει εξασθενημένη όραση.
- 1 στα 4 άτομα υποφέρει από πόνους στο χέρι, στον καρπό και στο βραχίονα του χεριού.
- 1 στα 5 άτομα έχει τουλάχιστον μία ελάχιστη απώλεια ακοής.<sup>[15]</sup>

## 1.4 Προσβασιμότητα

Η προσβασιμότητα είναι ένας γενικός όρος που χρησιμοποιείται για να περιγράψει το βαθμό κατά τον οποίο ένα προϊόν, μία συσκευή, μία υπηρεσία ή ένα περιβάλλον είναι προσβάσιμο από όσο το δυνατόν περισσότερα άτομα. Η προσβασιμότητα μπορεί να ειπωθεί ως η «δυνατότητα στην πρόσβαση» όπως κι ως πιθανό προνόμιο κάποιων συστημάτων ή οντοτήτων. Η προσβασιμότητα σαν όρος περισσότερο χρησιμοποιείται για να εστιάσει σε άτομα με αναπηρίες και το δικαίωμά για την πρόσβασή τους σε οντότητες, συχνά με τη βοήθεια βοηθητικών τεχνολογιών.<sup>[21]</sup>

Προσβασιμότητα σημαίνει δυνατότητα χρήσης μιας υπηρεσίας από τον καθένα, ανεξαρτήτως ιδιαιτερότητας (αναπηρία, ηλικία) ή πλαισίου χρήσης (context of use). Εντάσσεται σε ευρύτερο επιστημονικό πεδίο, της Αλληλεπίδρασης Ανθρώπου-Υπολογιστή (HCI) και ειδικότερα της Καθολικής Σχεδίασης (Universal Design) ή διαφορετικά Σχεδίασης για όλους (Design for all). Η προσβασιμότητα είναι εξ' ορισμού μια κατηγορία της ευχρηστίας. Οτιδήποτε δεν είναι προσβάσιμο σε ένα χρήστη δεν είναι και εύχρηστο.<sup>[6]</sup>

## 1.5 Σχεδίαση για όλους

**Σχεδίαση για Όλους** ή **Καθολική Σχεδίαση** σημαίνει μεγιστοποίηση της πιθανής αγοράς των προϊόντων ενός κατασκευαστή (παραγωγού) ή μιας εταιρείας μέσω της βεβαιότητας ότι ο μέγιστος αριθμός των ανθρώπων μπορεί να τα χρησιμοποιεί. Όσο περισσότεροι άνθρωποι μπορούν να χρησιμοποιούν ένα προϊόν, τόσο περισσότερα προϊόντα μπορούν να πουληθούν και τόσο μεγαλύτερα κέρδη να επιτευχθούν.

Μια πετυχημένη σχεδίαση για όλους εστιάζεται σε έναν κύριο ισχυρισμό, ότι το μοναδικό πιο σημαντικό συστατικό σε οποιοδήποτε σύστημα είναι ο χρήστης. Γνωρίζοντας

τους χρήστες, τις επιθυμίες τους, τις ανάγκες τους, τους πόθους τους και παρέχοντας σχεδιαστικές λύσεις που να ικανοποιούν αυτές τις επιθυμίες, ανάγκες και πόθους, ενισχύεται όχι μόνο η σχεδίαση για όλους, αλλά και η καλή σχεδίαση.

Παρ' όλα αυτά, οι περισσότεροι σχεδιαστές συστημάτων ποτέ δεν έχουν συναντήσει έστω και έναν από τους χρήστες των προϊόντων τους για να τον γνωρίσουν, να τον κατανοήσουν ή και να τον συμπαθήσουν.<sup>[22]</sup>

### 1.5.1 Καταρρίπτοντας τους μύθους της «σχεδίασης για όλους»

Η «σχεδίαση για όλους» δεν είναι μια νέα δραστηριότητα που πραγματοποιείται γιατί υπάρχει ένα θετικό ρεύμα σχετικά με αυτή. Δεν είναι μια προσέγγιση τέτοια, ώστε να καλύπτει μόνο τις "ειδικές ανάγκες" κάποιων ατόμων. Η "σχεδίαση για όλους" σχετίζεται με την απαίτηση να διασφαλιστεί ότι τα παραγόμενα προϊόντα έχουν την αποδοχή του μεγαλύτερου δυνατού αριθμού χρηστών. Μια διευρυμένη αγορά προσφέρει μεγαλύτερες πωλήσεις, που επιφέρουν και μεγαλύτερο κέρδος. Το κέρδος, η αύξηση των πωλήσεων και η εξασφάλιση ενός κομματιού της αγοράς αποτελούν βασικούς στόχους των εταιριών. Η σχεδίαση για όλους πρέπει επίσης να αποτελεί έναν από τους βασικούς στόχους τους.

Η σχεδίαση για όλους δεν είναι μια μεμονωμένη δραστηριότητα. Δεν είναι δυνατόν να διαχωριστεί από την βασική σχεδίαση και να αντιμετωπίζεται ως το τελευταίο χρονικά κομμάτι της διαδικασίας σχεδιασμού. Αντίθετα, πρέπει να θεωρείται βασική δραστηριότητα, ενταγμένη ολοκληρωτικά στην γενικότερη σχεδίαση.

Η σχεδίαση για όλους δεν είναι ο συγκεκριμένος τομέας μόνο κάποιων ειδικών. Όλοι οι σχεδιαστές οφείλουν να έχουν τα απαραίτητα προσόντα για να αντεπεξέλθουν σε αυτόν τον τομέα. Πρέπει απλά κάποιος να γνωρίζει τις σχετικές αρχές και να τις εφαρμόζει. Τέλος, η σχεδίαση για όλους δεν απευθύνεται μόνο στους αναπήρους και τους ηλικιωμένους.

Η σχεδίαση για όλους έχει σκοπό τη διεύρυνση των ορίων του σχεδιασμού. Επιδιώκει να απευθύνεται στο μεγαλύτερο δυνατό αριθμό χρηστών, συμπεριλαμβανομένων των υγιών ατόμων, των εξοικειωμένων με την τεχνολογία γενικά και των εργαζομένων, αλλά και των γονιών, των ηλικιωμένων ή ακόμα και κάποιου ο οποίος πολύ προσωρινά και για έναν ασήμαντο λόγο εντάχθηκε στην κατηγορία των αναπήρων (πχ ένα άτομο που προσπαθεί να διακρίνει το περιεχόμενο μίας οθόνης κινητού τηλεφώνου σε συνθήκες έντονης ηλιοφάνειας). Στόχος είναι να απαλλάσσει την καθημερινή ζωή από τα άχρηστα εμπόδια και να την κάνει καλύτερη και ευκολότερη για όλους.

Η προσοχή της σχεδίασης λογισμικού εστιάζεται περισσότερο στην αποδοχή, την ουσιαστική και την κοινωνική, καθώς αυτός είναι ο βασικός στόχος για τον σχεδιασμό προϊόντων και υπηρεσιών. Βασικό στοιχείο σε κάθε απόπειρα παραγωγής χρήσιμων και εύχρηστων προϊόντων είναι η κατανόηση των χρηστών, να βρεθούν δηλαδή οι ίδιοι οι σχεδιαστές συστημάτων στη θέση των αναπήρων και ηλικιωμένων, στη θέση των ανθρώπων που τους φροντίζουν, στη θέση του προσωπικού υποστήριξης ή να αισθανθούν όπως κάποιος που από απροσεξία βρέθηκε σε μια ανάλογη θέση.

Έχοντας αναγνωρίσει τους χρήστες και αποκτήσει γνώσεις για τις ανάγκες τους, τις επιθυμίες και τις φιλοδοξίες τους, γίνεται έπειτα η προσπάθεια να περιγραφούν τα χαρακτηριστικά τους και να βρεθούν μέθοδοι υπολογισμού του αριθμού των χρηστών με παρόμοια χαρακτηριστικά. Γίνεται ένα βήμα παραπάνω πέρα από την κατηγοριοποίηση σύμφωνα με την ηλικία ή την κατάσταση υγείας, και η προσοχή τώρα εστιάζεται στις



λειτουργικές τους ικανότητες - πόσο καλά μπορούν να πραγματοποιήσουν ενέργειες που είναι αναγκαίες, ώστε να υπάρξει αλληλεπίδραση με τα προϊόντα και τις υπηρεσίες.

Συνδυάζοντας την αναγνώριση των χρηστών με την κατανόηση της επιτακτικής ανάγκης για ανάπτυξη εξειδικευμένων προϊόντων, εισάγεται η έννοια της αφαίρεσης της σχεδίασης για όλους, με λίγα λόγια γίνεται προσπάθεια να αφαιρεθεί από τη γενικότερη σχεδίαση η εξειδίκευση. Το να αφαιρεθεί η εξειδίκευση, δηλαδή το να σχεδιαστεί ένα προϊόν ή μια υπηρεσία προσβάσιμη για τον καθένα, είναι μια ουτοπική ιδέα. Σε ένα πιο πρακτικό επίπεδο προτείνεται ότι η σχεδίαση για όλους θα πρέπει να στοχεύει στο να ελαχιστοποιήσει τον αποκλεισμό που προκύπτει από τις πρακτικές του «φτωχού» σχεδιασμού.

Ένα σημαντικό τμήμα της σχεδίασης για όλους πρέπει να είναι η δυνατότητα να εκτιμάται το επίπεδο ενσωμάτωσης (και αποκλεισμού) που επιτυγχάνεται καθώς αναπτύσσεται ο σχεδιασμός. Αυτό είναι ένα περίπλοκο και λεπτό ζήτημα, που συμπεριλαμβάνει την καταμέτρηση του επιπέδου λειτουργικών ικανοτήτων που απαιτούνται για να χρησιμοποιηθεί ένα προϊόν και τη χαρτογράφηση των χρηστών που δεν μπορούν να το χρησιμοποιήσουν. <sup>[22]</sup>

### 1.5.2 Η σημασία της σχεδίασης για όλους

Η σχεδίαση για όλους δεν είναι απλά μια προοπτική. Σύντομα θα μεταβληθεί σε επιχειρηματική αναγκαιότητα. Όλο και περισσότερες χώρες ψηφίζουν σχετική νομοθεσία, ώστε να εξασφαλίσουν ότι κανείς δεν υφίσταται διακρίσεις εξαιτίας της ηλικίας του ή των σωματικών ικανοτήτων του. Οι εταιρίες που δεν θα εναρμονιστούν με τις σχετικές επιταγές του νόμου, θα αντιμετωπίσουν πολυδάπανους δικαστικούς αγώνες.

Εκείνες οι εταιρίες που έχουν αγκαλιάσει τη σχεδίαση για όλους θα δρέψουν οικονομικά οφέλη. Τα προϊόντα που έχουν κατασκευαστεί με βάση τη φιλοσοφία της σχεδίασης για όλους, έχουν αυξημένα επίπεδα χρησιμότητας και προσβασιμότητας και οι καταναλωτές αντιδρούν πολύ θετικά σε αυτά. Οι εταιρίες αυτές κατασκευάζουν μια δυνατή επιχειρηματική εικόνα και εμπνέουν εμπιστοσύνη, σημειώνοντας ταυτόχρονα πρόοδο στον ανταγωνισμό με άλλες εταιρίες του αντίστοιχου κλάδου.

Καθώς παρουσιάζεται στους καταναλωτές ένας συνεχώς αυξανόμενος αριθμός χρήσιμων και προσβάσιμων προϊόντων, σταδιακά θα γυρίσουν τα απορρίπτουν τα προϊόντα που είναι δύσχρηστα. Υποστηρίζεται ότι καμία εταιρία δεν μπορεί να αγνοήσει μια τέτοια τάση. <sup>[22]</sup>

### 1.5.3 Αρχές καθολικής σχεδίασης

Οι αρχές της καθολικής σχεδίασης είναι οι εξής:

- Δικαίωμα στη χρήση
- Ευελιξία στη χρήση
- Απλότητα και διαισθητικότητα
- Αντιληπτικότητα πληροφοριών
- Ανοχή στα σφάλματα
- Χαμηλή φυσική προσπάθεια
- Κατάλληλο μέγεθος και χώρος για προσέγγιση και χρήση <sup>[22]</sup>

#### 1.5.4 Η καθολική πρόσβαση ή πρόσβαση για όλους

Η **Καθολική Πρόσβαση ή Πρόσβαση για Όλους (Universal Access ή Access for All)** είναι «το δικαίωμα όλων των πολιτών να έχουν και να διατηρούν πρόσβαση σε μια ευρεία κοινωνικά δεξαμενή πηγών πληροφόρησης και υπηρεσιών διαπροσωπικής επικοινωνίας, δεδομένης της ποικιλίας των πλαισίων χρήσης». Stephanidis C. (1998).<sup>[19]</sup>

Ο όρος **Καθολική Πρόσβαση ή Πρόσβαση για Όλους** χρησιμοποιείται και στον χώρο των Υποστηρικτικών Τεχνολογιών και αναφέρεται στην ανάπτυξη ειδικών διεπαφών χρήστη για υπολογιστές και άλλες τερματικές συσκευές που να είναι προσβάσιμες και χρηστικές από άτομα με χαμηλότερα επίπεδα ικανοτήτων.<sup>[22]</sup>

#### 1.6 Νομοθεσία

Όπως αναφέρθηκε και παραπάνω μια σειρά από χώρες ψηφίζουν τέτοια νομοθεσία, που εξασφαλίζει ότι κανείς δε θα υφίσταται διακρίσεις όσον αφορά την πρόσβασή του στην τεχνολογία εξαιτίας της ηλικίας του ή των σωματικών, πνευματικών και άλλων ιδιαιτεροτήτων και ικανοτήτων του, και μάλιστα ότι οι εταιρίες που δεν θα εναρμονιστούν με τις σχετικές επιταγές του νόμου, θα αντιμετωπίσουν πολυδάπανους δικαστικούς αγώνες. Ο κατάλογος αυτός των χωρών συνεχώς αυξάνεται.

Πιο συγκεκριμένα το 1998, με ειδικό νόμο το Αμερικάνικο κογκρέσο υποχρέωσε τις ομοσπονδιακές υπηρεσίες να κάνουν τις Τεχνολογίες Πληροφοριών και Επικοινωνίας (ΤΠΕ) προσβάσιμες σε ΑμεΑ (αναγνωρίζοντας ότι η απουσία προσβασιμότητας θέτει εμπόδια στην ικανότητα και το δικαίωμα του ατόμου να εντοπίζει και να λαμβάνει πληροφορία εύκολα και γρήγορα). Σύμφωνα με την παράγραφο 508 (Section 508) του νόμου, οι υπηρεσίες αυτές οφείλουν να παρέχουν σε υπαλλήλους και κοινό με αναπηρίες πληροφορία συγκρίσιμη με εκείνη στην οποία έχουν πρόσβαση όλοι οι υπόλοιποι.<sup>[17]</sup>

Παρόμοιες κινήσεις έχουν πραγματοποιηθεί και στην Ευρωπαϊκή Ένωση. Η Επιτροπή των Ευρωπαϊκών Κοινοτήτων (Commission of the European Communities) ορίζει πως το να «κάνει τα πλεονεκτήματα του τομέα της Τεχνολογίας των Πληροφοριών και των Επικοινωνιών (Information and Communication Technologies- ICT) διαθέσιμα στο όσο το δυνατόν μεγαλύτερο αριθμό ατόμων είναι κοινωνικά, ηθικά και πολιτικά αναγκαίο». <sup>[2]</sup> Στη βάση αυτή, οι επίσημοι ιστότοποι των θεσμών και των οργανισμών της Ευρωπαϊκής Ένωσης, υποχρεούνται να συμβαδίζουν και να πειθαρχούν με τις κατευθυντήριες οδηγίες και τα δεδομένα ανάπτυξης, όπως αυτά θέτονται από τους αντίστοιχους επίσημους διεθνείς οργανισμούς (Web Content Accessibility Guidelines- WCAG, World Wide Web Consortium- W3C ως μέρος του Web Accessibility Initiative- WAI).<sup>[1]</sup>

Πέρα από τη δημοκρατία που φαίνεται να προωθούν οι παραπάνω πολιτικές, είναι προφανές ότι δε λαμβάνουν ουσιαστικά υπ' όψιν τους βασικούς κοινωνικούς κι οικονομικούς παράγοντες. Τέτοιοι, είναι οι κυριότερες αιτίες που στερούν από το σύνολο σχεδόν του πληθυσμού (από τη συντριπτική πλειοψηφία στις αναπτυσσόμενες χώρες) την πρόσβαση στις σύγχρονες τεχνολογίες πληροφοριών κι επικοινωνίας. Είναι γεγονός πως οι άνεργοι κι οι άποροι λόγω οικονομικής δυσπραγίας, όσο κι αν απόλυτα πειθαρχήσουν οι εταιρίες με τους κανόνες προσβασιμότητας και σχεδίασης για όλους, δε θα έχουν πρόσβαση σε αυτού του είδους τις τεχνολογίες. Αυτόματα, υφίστανται έτσι έναν ακόμα κοινωνικό αποκλεισμό. Είναι χαρακτηριστικές άλλωστε οι εκτιμήσεις σύμφωνα με τις οποίες προσωπικό ηλεκτρονικό υπολογιστή διαθέτει ένας στους εκατό ανθρώπους στον κόσμο.<sup>[23]</sup>

## 1.7 Τελικά, γιατί προσβασιμότητα;

Μπορεί σε αυτό το σημείο να αναφερθούν σχηματικά κι ως συμπέρασμα οι παράγοντες οι οποίοι κάνουν αναγκαία τελικά την υιοθέτηση προτύπων κι οδηγιών προσβασιμότητας από τους σχεδιαστές συστημάτων και προϊόντων λογισμικού:

- Η ευαισθητοποίηση των σχεδιαστών και των αρμόδιων με τα θέματα ανάπτυξης τεχνολογίας πληροφοριών κι επικοινωνιών. Η μη τήρηση κανόνων κι οδηγιών προσβασιμότητας, αναπόφευκτα οδηγεί σε κοινωνικό αποκλεισμό μία μεγάλη μερίδα του πληθυσμού και σε καταστάσεις ανισότητας.
- Οι εταιρίες ανάπτυξης συστημάτων και προϊόντων λογισμικού (τουλάχιστον στις ανεπτυγμένες και σε αυτό τον τομέα χώρες), πλέον υποχρεούνται ως ένα βαθμό και νομοθετικά από τις πολιτείες να υιοθετούν τη λογική της «σχεδίασης για όλους». Είναι κι αυτός ένας λόγος άλλωστε που οι σχετικές εταιρίες σήμερα αναπτύσσουν (διαφημίζουν και προωθούν επίσης) αντίστοιχες πολιτικές.
- Η προσβασιμότητα κατακτά μεγαλύτερο μερίδιο στην αγορά. Το βελτιωμένα με τα πρότυπα της σχεδίασης για όλους προϊόντα, αναφέρονται σε σαφώς μεγαλύτερο ποσοστό καταναλωτών.
- Τα συγκεκριμένα συστήματα πλέον σε πλήθος περιπτώσεων, για να είναι συμβατά με τις νέες τεχνολογίες, πρέπει να υιοθετούν τα αντίστοιχα πρότυπα. Το δεδομένο αυτό είναι γνωστό ως πολυπραγμοσύνη (multi-modality).

# Κεφάλαιο 2: Εργαλεία υλοποίησης και παρουσίασης εφαρμογής

---

## 2.1 Αντικειμενοστραφής προγραμματισμός

Το μοντέλο αντικειμενοστραφούς προγραμματισμού έχει γίνει ιδιαίτερα δημοφιλές και είναι το ντε φάκτο πρότυπο για την ανάπτυξη λογισμικού σήμερα. Ένα αντικείμενο<sup>1</sup> είναι ένα μοντέλο λογισμικού, που περιέχει χαρακτηριστικά κατάστασης και συμπεριφοράς. Μέσα σε ένα πρόγραμμα, μπορούν να χρησιμοποιηθούν αντικείμενα για να απεικονιστεί οτιδήποτε.

Ο αντικειμενοστραφής προγραμματισμός<sup>2</sup>, εστιάζεται στην κατάσταση<sup>3</sup> και στη συμπεριφορά<sup>4</sup> διακριτών αντικειμένων. Αυτά τα αντικείμενα μπορούν να επικοινωνήσουν μεταξύ τους για να δημιουργήσουν την περίπλοκη λογική και λειτουργικότητα η οποία είναι αναγκαία στα περισσότερα σημερινά προγράμματα. Αυτή η μορφή επικοινωνίας αντικειμένου με αντικείμενο συχνά ονομάζεται μηνυματοδοσία. Η μηνυματοδοσία επιτυγχάνεται καλώντας τις μεθόδους<sup>5</sup> των αντικειμένων. Καθεμία από τις μεθόδους αυτές αλλάζει την εσωτερική κατάσταση ενός αντικειμένου (αν και αυτό δεν αποτελεί απαίτηση). Η κατάσταση ενός αντικειμένου παριστάνεται από τις μεταβλητές που ορίζονται μέσα στην ίδια κλάση<sup>6</sup>.

Τα αντικείμενα δημιουργούνται με βάση τις μεθόδους και τις μεταβλητές, που ορίζονται μέσα σε μια κλάση βάσης. Μια κλάση είναι σαν ένα κυανότυπο για τα αντικείμενα, και όλα τα αντικείμενα που χτίζονται από αυτή την κλάση, έχουν τα ίδια βασικά χαρακτηριστικά. Ωστόσο κάθε αντικείμενο διατηρεί τη δική του ιδιωτική κατάσταση. Δύο αντικείμενα που μοιράζονται τον ίδιο τύπο κλάσης είναι ανεξάρτητα μεταξύ τους σε ό,τι αφορά το θέμα της κατάστασης.<sup>[4]</sup>

---

<sup>1</sup> **Αντικείμενο:** Μια διακριτή μονάδα κώδικα στη μνήμη η οποία, όταν συνδυάζεται με άλλα αντικείμενα, μπορεί να δημιουργήσει πλήρεις εφαρμογές. Τα περισσότερα προγράμματα αποτελούνται από πολλαπλά αντικείμενα, τα οποία επικοινωνούν μεταξύ τους μέσω μεθόδων.

<sup>2</sup> **Αντικειμενοστραφής προγραμματισμός:** Μια μεθοδολογία προγραμματισμού, που οργανώνει προγράμματα ακολουθώντας το μοντέλο του πραγματικού κόσμου. Στον πραγματικό κόσμο, τα αντικείμενα συχνά αποτελούνται από μικρότερα συστατικά. Στην αντικειμενοστραφή ανάπτυξη εφαρμόζεται η ίδια αρχή, η οποία οδηγεί σε ευέλικτο, επαναχρησιμοποιήσιμο κώδικα.

<sup>3</sup> **Κατάσταση:** Τα δεδομένα ενός προγράμματος ή μιας εφαρμογής. Για παράδειγμα το υπόλοιπο ενός τραπεζικού λογαριασμού είναι μια κατάσταση. Επόμενες λειτουργίες μπορούν να λειτουργήσουν επί αυτής της κατάστασης. Για παράδειγμα, μια μέθοδος μπορεί να υπολογίσει τον τόκο του υπολοίπου, το οποίο μεταβιβάζεται στη μέθοδο.

<sup>4</sup> **Συμπεριφορά:** Η συλλογή μεθόδων για μια συγκεκριμένη κλάση. Η συμπεριφορά των αντικειμένων χειρίζεται την κατάσταση των αντικειμένων.

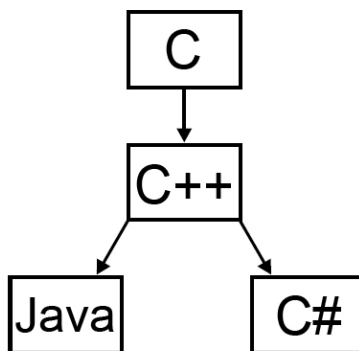
<sup>5</sup> **Μέθοδος:** Μια μονάδα κώδικα που εκτελεί μία ή περισσότερες ενέργειες. Για παράδειγμα, ένα αντικείμενο μπορεί να έχει μια μέθοδο με όνομα print, η οποία στέλνει ένα έγγραφο στον εκτυπωτή. Σε άλλες γλώσσες (πέραν της C# που θα αναφερθεί παρακάτω), οι μέθοδοι μερικές φορές καλούνται συναρτήσεις, διαδικασίες και πράξεις.

<sup>6</sup> **Κλάση:** Θεμελιώδες συστατικό των γλωσσών αντικειμενοστραφούς προγραμματισμού. Μια κλάση είναι ένα πρότυπο για έναν τύπο που ορίζεται από το χρήστη. Από μια μόνο κλάση μπορούν να δημιουργηθούν αρκετά αντικείμενα.

## 2.2 Η γλώσσα προγραμματισμού Microsoft Visual C#

Η C# γενικά ως γλώσσα προγραμματισμού, έχει στενή συγγένεια με τις αντίστοιχες C, C++ και Java. Αυτό δε συνέβη τυχαία. Οι τρεις αυτές γλώσσες είναι οι πιο ευρέως χρησιμοποιημένες- και πιο ευρέως συμπαθείς- γλώσσες προγραμματισμού στον κόσμο της πληροφορικής. Επιπλέον, σχεδόν όλοι οι επαγγελματίες προγραμματιστές έως τις μέρες μας γνώριζαν λίγο ως πολύ τις παραπάνω γλώσσες. Ωστόσο, η C# δεν αναπτύχθηκε για να «επανεφευρεθεί ο τροχός», αλλά για να δοθεί μία ελευθερία στο να εστιαστεί η προσοχή σε κάποιες συγκεκριμένες βελτιώσεις και καινοτομίες.

Το «οικογενειακό δέντρο» της C# παρουσιάζεται παρακάτω στο σχήμα 2-1. Ο «παππούς» της C# είναι η C. Από τη C, η C# κληρονομεί τη σύνταξή της, πολλές από τις λέξεις- κλειδιά της και τους χειριστές της. Η C# βασίζεται πάνω στο μοντέλο αντικειμενοστραφούς προγραμματισμού που ορίζεται από τη C++, βεβαίως όμως βελτιώνοντάς το.



Σχήμα 2-1: Το Οικογενειακό Δέντρο της C#<sup>[10]</sup>

Η γλώσσα προγραμματισμού Microsoft Visual C# είναι μια πανίσχυρη αλλά και εύκολη γλώσσα προγραμματισμού, προσανατολισμένη κυρίως σε προγραμματιστές που δημιουργούν εφαρμογές με το Πλαίσιο Εφαρμογών Microsoft .NET Framework. Έχει κληρονομήσει πολλές από τις πιο καλές λειτουργίες της C++ και της Microsoft Visual Basic, αλλά πολύ λίγες από τις ασυνέπειες και τους αναχρονισμούς τους, με αποτέλεσμα μια πιο σαφή και λογική γλώσσα.<sup>[11]</sup>

Η Microsoft C# είναι γλώσσα προσανατολισμένη σε συστατικά (component- oriented). Ο σημαντικός ρόλος, όπως προαναφέρθηκε που παίζει στην αρχιτεκτονική του Πλαισίου (Ανάπτυξης) Εφαρμογών .NET της Microsoft, μερικοί τον συγκρίνουν με το ρόλο που έπαιξε η C στην ανάπτυξη του UNIX.<sup>[11]</sup> Η C# είναι σήμερα μία πλήρως ώριμη γλώσσα, που έχει σαν βάση της την πείρα που έχει συσσωρευτεί στον προγραμματισμό για πάνω από τρεις δεκαετίες.<sup>[5]</sup>

### 2.2.1 Η έκδοση Microsoft Visual C# 3.0

Στόχος της C# 3.0 είναι να προβάλλει μία απλή, ασφαλή, μοντέρνα, προσανατολισμένη σε αντικείμενα (αντικειμενοστραφή) και σε ανάπτυξη εφαρμογών για τον παγκόσμιο ιστό, υψηλής απόδοσης γλώσσα για την ανάπτυξη με το .NET.<sup>[5]</sup> Με την έκδοση 3.0, η Microsoft έβαλε τη C# στην πιο εξέχουσα θέση του γλωσσικού σχεδιασμού, προσθέτοντας ένα

σύνολο από καινοτόμα χαρακτηριστικά γνωρίσματα που επαναπροσδιόρισαν το τοπίο του προγραμματισμού.

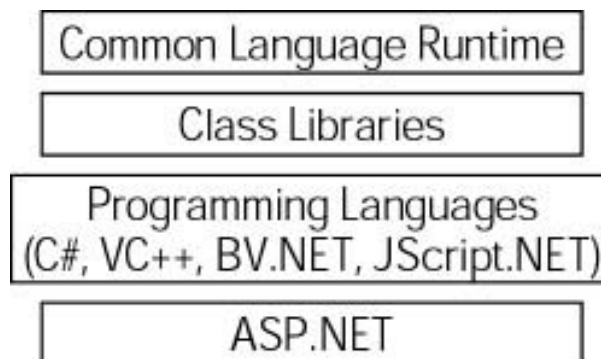
Ίσως τα δύο πιο συναρπαστικά νέα χαρακτηριστικά γνωρίσματα στη C# 3.0 είναι οι γλωσσικές ενσωματωμένες ερωτήσεις (LINQ) και οι εκφράσεις lambda.LINQ που επιτρέπουν γραφή ερωτήσεων προς βάσεις δεδομένων χρησιμοποιώντας τα στοιχεία προγραμματισμού της C#. Οι εκφράσεις lambda χρησιμοποιούνται συχνά στις εκφράσεις LINQ. Μαζί προσθέτουν μια εξ ολοκλήρου νέα διάσταση στον προγραμματισμό της C#. <sup>[10]</sup> Άλλα νέα χαρακτηριστικά περιλαμβάνουν:

- Εκφράσεις lambda (ανώνυμες εντολές σε στεροειδή).
- Μέθοδοι επέκτασης.
- Αρχικοποιήσεις αντικειμένων (τοποθέτηση αρχικών τιμών σε αντικείμενα).
- Ανώνυμοι τύποι.
- Σιωπηρά δακτυλογραφημένες τοπικές μεταβλητές.
- Σιωπηρά δακτυλογραφημένοι πίνακες.
- Δέντρα εκφράσεων.
- Αυτόματες ιδιότητες. <sup>[5]</sup>

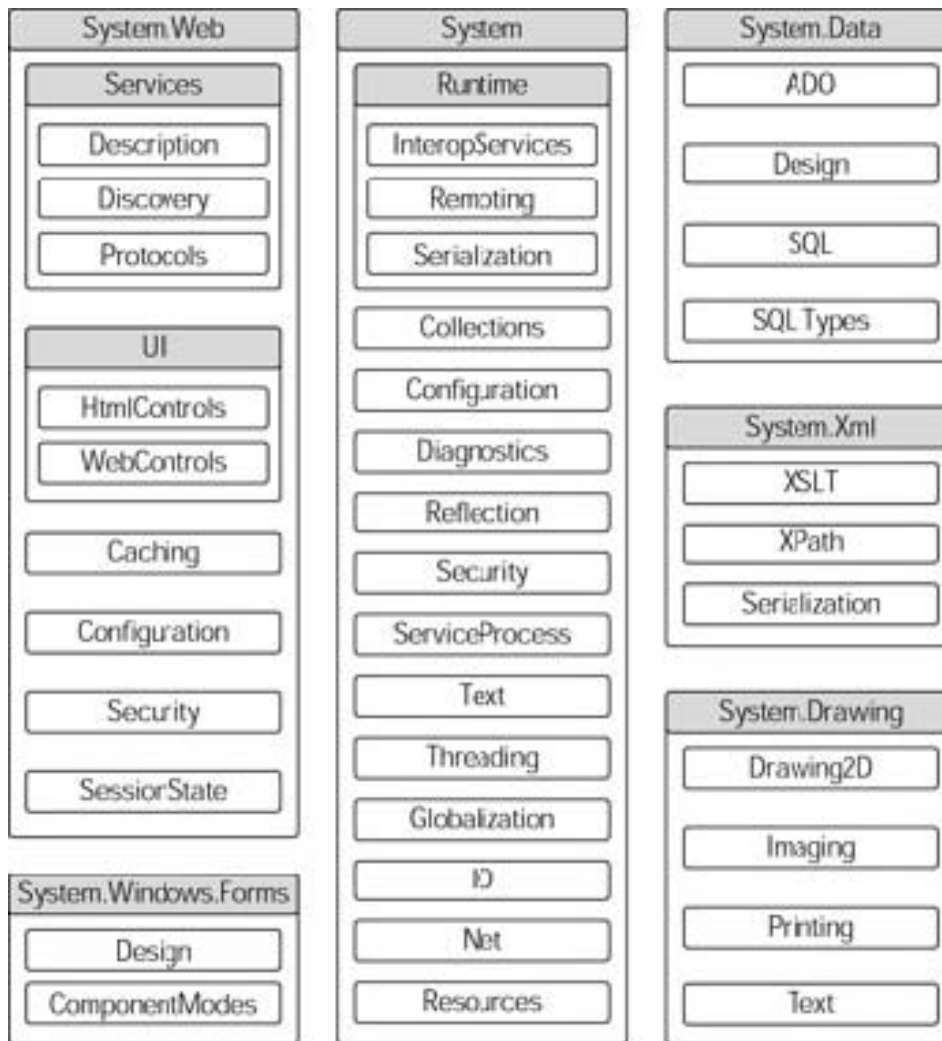
### 2.3 Το πλαίσιο ανάπτυξης εφαρμογών Microsoft .NET Framework

Με μία πρόταση, το πλαίσιο εφαρμογών Microsoft .NET Framework ορίζει ένα περιβάλλον που υποστηρίζει την ανάπτυξη και την εκτέλεση υψηλά κατανεμημένων και βασισμένων σε συστατικά εφαρμογών. Παρέχει τη δυνατότητα της ταυτόχρονης χρήσης διαφορετικών γλωσσών προγραμματισμού και εξασφαλίζει την ασφάλεια, τη μεταφερισιμότητα του προγράμματος κι ένα κοινό προγραμματιστικό μοντέλο για την πλατφόρμα των Windows. <sup>[10]</sup>

Το πλαίσιο εφαρμογών .NET είναι οργανωμένο σε τέσσερα μέρη όπως φαίνεται και στο σχήμα 2-2:



**Σχήμα 2-2:** Τα τέσσερα συστατικά του πλαισίου εφαρμογών .NET <sup>[3]</sup>



**Σχήμα 2-3:** Οι Βιβλιοθήκες Κλάσεων του πλαισίου εφαρμογών .NET<sup>[3]</sup>

- το Χρόνο Εκτέλεσης Κοινής Γλώσσας (*Common Language Runtime*).<sup>[3]</sup> Αυτό είναι ένα σύστημα που διευθύνει την εκτέλεση του προγράμματος. Μαζί και με άλλα πλεονεκτήματα, ο Χρόνος Εκτέλεσης Κοινής Γλώσσας είναι ένα μέρος του πλαισίου εφαρμογών .NET που καθιστά τα προγράμματα ώστε να είναι φορητά, υποστηρίζει τον προγραμματισμό με περισσότερες από μία γλώσσες και εξασφαλίζει την ασφάλεια.<sup>[10]</sup>
- Ένα σύνολο από *Βιβλιοθήκες Κλάσεων*<sup>7</sup> (*Class Libraries*). Αυτή η βιβλιοθήκη δίνει στο πρόγραμμα πρόσβαση στο περιβάλλον του χρόνου εκτέλεσης. Για παράδειγμα, αν θέλουμε να φτιάξουμε μια εφαρμογή τύπου I/O<sup>8</sup>, όπως το να εμφανίζεται κάτι στην οθόνη, θα χρησιμοποιηθεί το πλαίσιο εφαρμογών .NET για να γίνει. Από τη στιγμή που το πρόγραμμα περιορίζεται στα χαρακτηριστικά που καθορίζονται από τη βιβλιοθήκη κλάσεων του .NET, τα προγράμματα μπορούν να εκτελεστούν οπουδήποτε το σύστημα χρόνου εκτέλεσης του .NET υποστηρίζεται. Από τη στιγμή που η γλώσσα προγραμματισμού C# χρησιμοποιεί αυτόματα τη βιβλιοθήκη

<sup>7</sup> **Κλάση:** Η κλάση είναι μία αντικειμενοστραφής κατασκευή που βοηθάει στην οργάνωση των προγραμμάτων.<sup>[10]</sup>

<sup>8</sup> **I/O (Input/ Output) (= Είσοδος/ Έξοδος (Πληροφοριών):** Η συλλογή των διεπαφών όπου οι διαφορετικές λειτουργικές μονάδες ενός συστήματος επεξεργασίας πληροφοριών χρησιμοποιούν για να επικοινωνήσουν η μία με την άλλη.<sup>[21]</sup>



κλάσεων του .NET, τα προγράμματα που έχουν δημιουργηθεί σε C# γλώσσα είναι αυτόματα μεταφέρσιμα σε όλα τα περιβάλλοντα του .NET. <sup>[10]</sup>

- Ένα σύνολο από γλώσσες προγραμματισμού και
- Το περιβάλλον του *ASP.NET*. <sup>[3]</sup>

Το πλαίσιο εφαρμογών .NET σχεδιάστηκε για να εκπληρώσει τρεις κύριους στόχους: Πρώτον, σχεδιάστηκε για να κάνει τις εφαρμογές των Windows πολύ περισσότερο αξιόπιστες, καθώς ταυτόχρονα θα εξασφάλιζε μια εφαρμογή με έναν υψηλότερο βαθμό ασφάλειας. Δεύτερον, σχεδιάστηκε για να απλουστεύσει την ανάπτυξη WEB εφαρμογών και υπηρεσιών που λειτουργούν μόνο με τον παραδοσιακό τρόπο (εφαρμογές desktop), αλλά και σε συσκευές κινητών τηλεφώνων. Τέλος, το πλαίσιο αυτό σχεδιάστηκε για να εφοδιάσει τον προγραμματιστή με ένα σύνολο από βιβλιοθήκες που θα λειτουργούν με πολλαπλές γλώσσες (βλ. Σχήμα 2-3). <sup>[3]</sup>

### 2.3.1 Τα πλεονεκτήματα του πλαισίου ανάπτυξης εφαρμογών Microsoft .NET

Συνοπτικά, μερικά από τα χαρακτηριστικά του πλαισίου εφαρμογών .NET είναι τα εξής:

- **Ο αντικειμενοστραφής προγραμματισμός** – Και το πλαίσιο εφαρμογών .NET και η γλώσσα προγραμματισμού C# είναι ολοκληρωτικά βασισμένα στις αρχές του αντικειμενοστραφούς προγραμματισμού.
- **Καλός σχεδιασμός** – Μια βασική βιβλιοθήκη κλάσης, η οποία έχει σχεδιαστεί από τη βάση με έναν υψηλά διαισθητικό τρόπο.
- **Ανεξαρτησία γλώσσας** – Με το .NET, όλες οι γλώσσες Visual Basic .NET, C#, J# κι η διαχειριζόμενη C++ μεταγλωττίζονται σε μία κοινή **Ενδιάμεση Γλώσσα**. Αυτό σημαίνει ότι οι γλώσσες γίνονται διαλειτουργικές μ' έναν τρόπο που συναντάται στο παρελθόν.
- **Καλύτερη υποστήριξη για δυναμικές ιστοσελίδες** – Ενώ η τεχνολογία ASP προσέφερε ιδιαίτερη ευκαμψία, ήταν ταυτόχρονα αναποτελεσματική εξαιτίας της χρήσης ερμηνευμένης γλώσσας σεναρίου και της έλλειψης αντικειμενοστραφούς σχεδιασμού, που συχνά οδηγούσε σε ένα μπερδεμένο ASP κώδικα. Το .NET προσφέρει μια ολοκληρωμένη υποστήριξη για τις ιστοσελίδες, χρησιμοποιώντας μια νέα τεχνολογία – την ASP.NET. Με την ASP.NET, ο κώδικας στις σελίδες είναι μεταγλωττισμένος και μάλιστα μπορεί να είναι γραμμένος στο .NET – γνωρίζοντας γλώσσες υψηλού επιπέδου όπως η C#, η J# ή η Visual Basic .NET.
- **Αποδοτική πρόσβαση δεδομένων** – Ένα σύνολο .NET συστατικών, γνωστό ως ADO.NET, που εξασφαλίζει αποδοτική πρόσβαση σε σχεσιακές βάσεις δεδομένων και σε μία ποικιλία από πηγές δεδομένων. Τα συστατικά είναι επίσης διαθέσιμα για να επιτρέπουν την πρόσβαση στο σύστημα αρχείων και στα λεξικά. Συγκεκριμένα, η υποστήριξη της XML είναι χτισμένη πάνω στο .NET, επιτρέποντας τη διαχείριση δεδομένων, τα οποία μπορεί να εισαχθούν ή να εισαχθούν σε πλατφόρμες πέρα των Windows.
- **Καταμερισμός κώδικα** – Το .NET έχει ολοκληρωτικά ξαναδημιουργήσει τον τρόπο με τον οποίο ο κώδικας διαμοιράζεται (καταμερίζεται) μεταξύ εφαρμογών, εισάγοντας την αρχή της **συναρμολόγησης (assembly<sup>9</sup>)**, η οποία αντικαθιστά τον

<sup>9</sup> **Assembly (Συναρμολόγηση):** Στο .NET πλαίσιο εφαρμογών, μια συναρμολόγηση είναι μια μεταγλωττισμένη βιβλιοθήκη κώδικα για τη χρήση στην επέκταση, για την έκδοση τύπου αρχείων και την ασφάλεια. Υπάρχουν δύο ειδών συναρμολογήσεις- οι συναρμολογήσεις διαδικασίας (.EXE) κι οι συναρμολογήσεις βιβλιοθήκης (.DLL). <sup>[21]</sup>



παραδοσιακό τρόπο με το DLL<sup>10</sup>. Οι συναρμολογήσεις έχουν τυπικές ευκολίες για την εκδοτική (τύπου αρχείων) και διαφορετικές εκδόσεις συναρμολογήσεων μπορούν να υπάρχουν ταυτόχρονα.

- **Βελτιωμένη ασφάλεια** – Κάθε συναρμολόγηση μπορεί να διαθέτει ενσωματωμένη ασφάλεια πληροφοριών η οποία μπορεί να υποδεικνύει επακριβώς ποιος ή τι κατηγορία χρηστών ή διαδικασίας επιτρέπεται να καλέσουν ποιες μεθόδους σε ποιες κλάσεις. Αυτό προσφέρει έναν πολύ ικανοποιητικό βαθμό ελέγχου στο πως οι συναρμολογήσεις που αναπτύσσονται μπορούν να χρησιμοποιηθούν.
- **Μηδενικός αντίκτυπος εγκατάστασης** – Υπάρχουν δύο τύποι συναρμολογήσεων: οι κατανεμημένες κι οι ιδιωτικές. Οι κατανεμημένες συναρμολογήσεις είναι κοινές βιβλιοθήκες διαθέσιμες σε όλα το λογισμικό (γνωστές κι ως DLL), ενώ οι ιδιωτικές προορίζονται για χρήση μόνο από ένα συγκεκριμένο λογισμικό. Μία ιδιωτική συναρμολόγηση είναι ολοκληρωτικά ανεξάρτητη, ώστε η διαδικασία της εγκατάστασης να είναι απλή. Δεν υπάρχουν εγγραφές εισόδου: τα κατάλληλα αρχεία, τοποθετούνται στους κατάλληλους φακέλους μέσα στο σύστημα αρχείων.
- **Υποστήριξη για τις υπηρεσίες του Web** – Το .NET έχει πλήρως ολοκληρωμένη υποστήριξη για την ανάπτυξη υπηρεσιών WEB ώστε αυτές να αναπτύσσονται το ίδιο εύκολα όπως κι οποιαδήποτε άλλος τύπος εφαρμογής.
- **Visual Studio .NET** – Το .NET ήρθε με ένα περιβάλλον ανάπτυξης, το Visual Studio .NET, το οποίο μπορεί να ταιριάζει το ίδιο καλά με τη C++, τη C#, τη J#, τη Visual Basic .NET όπως και με τον ASP.NET κώδικα. Το Visual Studio .NET συνενώνει σε ένα ενιαίο σύνολο όλα τα καλύτερα χαρακτηριστικά των αντίστοιχων περιβαλλόντων συγκεκριμένων γλωσσών, των Visual Studio 6.
- **C#** - Η C# είναι μια νέα αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε για χρήση σε συνεργασία με το .NET.<sup>[3]</sup>

### 2.3.2 Η έκδοση .NET Framework 3.5

Η έκδοση 3.5 του πλαισίου ανάπτυξης εφαρμογών .NET αντιπροσωπεύει μία περαιτέρω ωρίμανσή του και φέρνει μαζί της νέους τρόπους για τη αξιολογή δημιουργία όλων των ειδών λογισμικού, χωρίς να μετατρέπει σε ξεπερασμένη γνώση οτιδήποτε οι μηχανικοί λογισμικού έχουν μάθει, τουλάχιστον γύρω από το .NET, έως σήμερα.

Υπάρχει ακόμη η δυνατότητα για τη δημιουργία WEB εφαρμογών που «τρέχουν» από την πλευρά του server, αλλά αυτή τη φορά αναπτύσσονται και με τη βοήθεια της τεχνολογίας AJAX, όπως μπορούν επίσης να προστεθούν ρυθμίσεις από την πλευρά του πελάτη (η τεχνολογία AJAX παρέχει υποστήριξη για πολύ περισσότερες δυνατότητες, συμπεριλαμβανομένης της JSON<sup>11</sup> κωδικοποίησης και αποκωδικοποίησης). Ακόμη, μπορούν να δημιουργηθούν Windows Forms εφαρμογές (πχ εφαρμογές desktop για το

---

<sup>10</sup> **Dynamic Link Library (Βιβλιοθήκη Δυναμικών Συνδέσεων)**: Η Βιβλιοθήκη Δυναμικών Συνδέσεων ή DLL, είναι μια εφαρμογή της Microsoft με την κοινή έννοια των βιβλιοθηκών για τα λειτουργικά συστήματα Windows και OS/2. Σύμφωνα με αυτή,, οποιοδήποτε αρχείο στοιχείων με την ίδια μορφή αρχείου, μπορεί να κληθεί ως πόρος DLL.<sup>[21]</sup>

<sup>11</sup> **JSON- JavaScript Object Notation (=Σημείωση Αντικειμένου για τη JavaScript)**: Είναι ένα ελαφρύ βασισμένο στο κείμενο ανοιχτό πρότυπο που σχεδιάζεται για την κατανοήσιμη από τον άνθρωπο ανταλλαγή στοιχείων. Προέρχεται από τη γλώσσα προγραμματισμού JavaScript για την αντιπροσώπευση των απλών δομών δεδομένων και των συνειρμικών σειρών, αποκαλούμενων ως αντικείμενα. Παρά τη σχέση του σε JavaScript, είναι ανεξάρτητο από τη γλώσσα (=language- independent), με τους καταμητές να είναι διαθέσιμοι για ουσιαστικά κάθε γλώσσα προγραμματισμού.<sup>[21]</sup>

λειτουργικό σύστημα των WINDOWS) για τα ίδια τα Windows, αλλά ακόμη καλύτερα, μπορούν να δημιουργηθούν πλουσιότερες Windows εφαρμογές με τη χρήση της WPF τεχνολογίας (Windows Presentation Foundation – Βάση Παρουσίασης των Windows), η οποία χρησιμοποιεί μια επεξηγηματική τεχνολογία σύνταξης που ονομάζεται XAML<sup>12</sup> (Extensible Application Markup Language). Η XAML επίσης χρησιμοποιείται στην ανάπτυξη κλασικών Windows Forms εφαρμογών, οι οποίες μπορούν να φανούν χρήσιμες, πέραν των άλλων, ως επιχειρησιακό στρώμα για τις εφαρμογές.

Σε μία από τις πιο εντυπωσιακές προσθήκες της συγκεκριμένης έκδοσης του πλαισίου ανάπτυξης, υπάρχει η δυνατότητα ο ίδιος κώδικας XAML μπορεί να παράγει cross-platform (as of writing, Windows, Mac, Unix) και cross-browser (Firefox και Safari), πλούσιες εφαρμογές για το INTERNET με τη χρήση του λογισμικού Microsoft Silverlight<sup>13</sup>.

Όλες αυτές οι τεχνολογίες για την ανάπτυξη λογισμικού μπορούν να χρησιμοποιήσουν τη γλώσσα C# για την προγραμματιστική λογική: η C# μπορεί να αποτελέσει τη βάση για όλο τον προγραμματισμό που πραγματοποιείται επί των ορίων της ανάπτυξης .NET εφαρμογών από το WEB έως το desktop (=επιφάνεια εργασίας), από τους αρχάριους μηχανικούς λογισμικού έως τους επαγγελματίες κι από πλούσιες INTERNET εφαρμογές έως WEB υπηρεσίες.<sup>[5]</sup>

## 2.4 Η σχέση μεταξύ Microsoft .NET και Microsoft Visual C#

Αν και η C# είναι μια γλώσσα προγραμματισμού που μπορεί να μελετηθεί από μόνη της, έχει ωστόσο μια ειδική σχέση με το περιβάλλον χρόνου εκτέλεσής της, το πλαίσιο ανάπτυξη εφαρμογών .NET. Ο λόγος για τον οποίο ισχύει το παραπάνω γεγονός είναι διττός. Πρώτον, η C# είχε αρχικά σχεδιαστεί από τη Microsoft για να δημιουργήσει η τελευταία κώδικα για το .NET πλαίσιο. Δεύτερον, οι βιβλιοθήκες που χρησιμοποιούνται από τη C# είναι αυτές που καθορίζονται από το .NET. Γι' αυτό το λόγο, έστω κι αν είναι δυνατό να αυτονομήσουμε τη C# γλώσσα από το .NET περιβάλλον, τα δυο τους είναι στενά συνδεδεμένα. Εξ' αιτίας αυτού, είναι σημαντικό κάποιος να έχει μια γενική γνώση για το τι είναι το πλαίσιο το .NET και γιατί αυτό συνδέεται με τη C# γλώσσα προκειμένου να προγραμματίσει με την τελευταία.<sup>[10]</sup>

Η C# ωστόσο δεν είναι από μόνη της μέρος του .NET. Υπάρχουν κάποια χαρακτηριστικά που υποστηρίζονται από το .NET αλλά όχι από τη C# κι είναι πιθανό κάποιος να εκπλαγεί από το γεγονός να μάθει ότι υπάρχουν στην πραγματικότητα κάποια χαρακτηριστικά της γλώσσας C# τα οποία δεν υποστηρίζονται από το .NET (για παράδειγμα, κάποιες υποδείξεις του τελεστή υπερφόρτωσης).<sup>[9]</sup>

## 2.5 Λογισμικό ανάπτυξης εφαρμογών Microsoft Visual Studio 2008

Το Visual Studio είναι ένα προϊόν λογισμικού που ενσωματώνει τις τελευταίες προόδους των γλωσσών προγραμματισμού της Microsoft, της Visual Basic και της C#,

---

<sup>12</sup> **XAML- Extensible Application Markup Language (= Επεκτάσιμη Γλώσσα Σήμανσης για Εφαρμογές):** Είναι μια βασισμένη στην XML γλώσσα σήμανσης που δημιουργήθηκε από τη Microsoft και χρησιμοποιείται για να χαρακτηρίζει μονοσήμαντα τις δομημένες τιμές και τα αντικείμενα.<sup>[21]</sup>

<sup>13</sup> **Microsoft Silverlight:** Ένα πλαίσιο εφαρμογής Ιστού που ενσωματώνει πολυμέσα, ηλεκτρονική γραφιστική, ζωτικότητα (animation) και αλληλεπίδραση σε ένα ενιαίο περιβάλλον χρόνου εκτέλεσης. Αρχικά εκδόθηκε για τη συνεχή σύνδεση βίντεο (video streaming plug-in), οι πιο πρόσφατες όμως εκδόσεις έφεραν πρόσθετα χαρακτηριστικά γνωρίσματα και την υποστήριξη αλληλεπίδρασης για τις γλώσσες και τα εργαλεία ανάπτυξης τύπου CLI.<sup>[21]</sup>

ταυτόχρονα με ένα πλήθος από βελτιώσεις και νέα χαρακτηριστικά στην αλληλεπίδραση με το χρήστη. Απευθύνεται σε νέους με το εργαλείο .NET σχεδιαστές λογισμικού αλλά και σε πιο έμπειρους.

Η έκδοση Professional Visual Studio 2008, εξετάζει κάθε σημαντική πτυχή αυτού του εργαλείου ανάπτυξης λογισμικού, δείχνοντας στο χρήστη πώς να χρησιμοποιεί κάθε χαρακτηριστικό του και προτείνοντάς του συμβουλές για το πώς καλύτερα θα αξιοποιήσει τα διάφορα συστατικά του αποτελεσματικότερα. Παρουσιάζει επίσης τις δομικές μονάδες που αποτελούν το Visual Studio 2008, χωρίζοντας τη διεπαφή με τον χρήστη σε εύχρηστα μεγάλα κομμάτια για την καλύτερη κατανόηση του προγράμματος. Επεκτείνεται έπειτα σε κάθε ένα από αυτά τα συστατικά με πρόσθετες λεπτομέρειες για το πώς ακριβώς λειτουργεί και ανεξάρτητα αλλά και από κοινού με άλλα μέρη του Visual Studio για να καταστήσει τις προσπάθειες ανάπτυξης λογισμικού αποδοτικότερες.

Όταν το Visual Studio 2005 και το .NET 2.0 πλαίσιο εκδόθηκαν, ήταν δυο σημαντικές νέες εκδόσεις με νέες θεμελιώδεις κλάσεις πλαισίου που επεκτάθηκαν αρκετά πέρα από όσα η Microsoft είχε αναπτύξει έως τότε. Εντούτοις, το περισσότερο σημαντικό μέρος αυτών των εκδόσεων πραγματοποιήθηκε στο IDE<sup>14</sup> όπου τα διάφορα συστατικά ταίριαξαν μαζί με έναν συνεκτικό τρόπο, ώστε να παράσχουν ένα αποδοτικό σύνολο εργαλείων για να καταστήσουν τα πάντα μέσα στο πρόγραμμα και στο πλαίσιο ευπρόσιτα.

Η πιο πρόσφατη έκδοση όμως, η Visual Studio 2008 και το πλαίσιο .NET 3.5, στηρίζεται στο ισχυρό θεμέλιο που ονομάζεται .LINQ<sup>15</sup> το οποίο ουσιαστικά άλλαξε τον τρόπο πρόσβασης στα δεδομένα και τα χαρακτηριστικά γνωρίσματα που ήταν προηγουμένως διαφορετικά φορτωμένα, όπως το ASP.NET, η τεχνολογία AJAX και τα εργαλεία του Visual Studio για τα Office. Όλα αυτά πλέον συμπεριλαμβάνονται στο πρόγραμμα εξ' ορισμού (by default).

Το περιβάλλον ανάπτυξης Visual Studio 2008 (βλ. σχήμα 2 - 4) πήγε την εξέλιξη των Microsoft IDEs ακόμα παραπέρα δημιουργώντας ένα περιεκτικό σύνολο εργαλείων που μπορεί να χρησιμοποιηθεί ανεξάρτητα από το σκοπό κάποιου ως υπεύθυνου για την ανάπτυξη προϊόντων λογισμικού. Το σχήμα 2 - 4 παρουσιάζει ένα συνεκτικό τρόπο με τον οποίο τα διάφορα συστατικά οργανώνονται κατάλληλα μεταξύ τους μέσα στο πρόγραμμα ώστε να παρέχουν στο χρήστη ένα αποδοτικό σύνολο εργαλείων όπου όλα είναι ευπρόσιτα.<sup>[8]</sup>

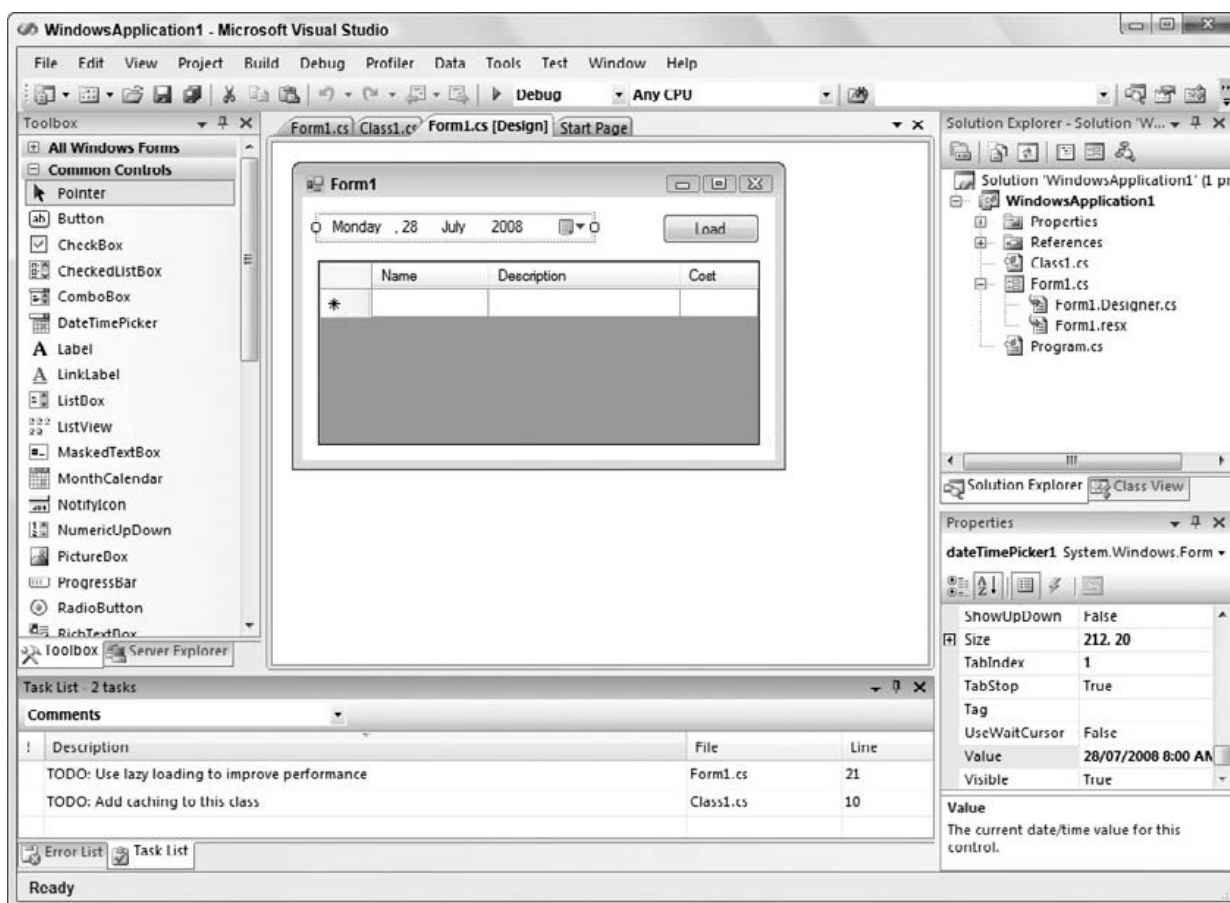
Το Visual Studio 2008 αντιπροσωπεύει έναν επανασυγχρονισμό των εργαλείων ανάπτυξης και του .NET πλαισίου. Το .NET πλαίσιο εξελίσσεται από την έκδοση 3.0 (εκδόθηκε μαζί με την έκδοση Vista των Windows) στην 3.5, μια μετεξελιγμένη έκδοση που «δένεται» μαζί με το Visual Studio 2008. Το IDE τώρα εσωτερικά υποστηρίζει εφαρμογές τύπου Windows Presentation Foundation (WPF), Windows Communication Foundation (WCF) και Windows Workflow Foundation (WWF). Επιπλέον, υπάρχουν και

---

<sup>14</sup> **IDE- Integrated Development Environment (= Ολοκληρωμένο (ή Ενσωματωμένο) Περιβάλλον Ανάπτυξης):** Το Ολοκληρωμένο Περιβάλλον Ανάπτυξης, γνωστό κι ως Ολοκληρωμένο Περιβάλλον Σχεδιασμού ή Ολοκληρωμένο Περιβάλλον Εκσφαλμάτωσης είναι μια εφαρμογή λογισμικού που περιέχει περιεκτικές ευκολίες στους προγραμματιστές υπολογιστών για ανάπτυξη λογισμικού. Το IDE κανονικά αποτελείται από: ένα συντάκτη πηγαίου κώδικα, ένα μεταγλωττιστή ή/ και ένα διερμηνέα, εργαλεία αυτοματοποιημένης ανάπτυξης και ένα διορθωτή (debugger).<sup>[21]</sup>

<sup>15</sup> **LINQ- Language Integrated Query (= Ολοκληρωμένη (ή Ενσωματωμένη) Γλώσσα Ερώτησης):** Είναι ένα τμήμα του πλαισίου Microsoft .NET που προσθέτει εγγενή δεδομένα για την ικανότητα πραγματοποίησης ερωτήσεων (για τη λήψη στοιχείων- δεδομένων) προς τις γλώσσες προγραμματισμού του .NET.<sup>[21]</sup>

πολλές άλλες νέες εφαρμογές και δυνατότητες, συμπεριλαμβανομένων την άμεση υποστήριξη εφαρμογών.



Εικόνα 2-1: Το περιβάλλον ανάπτυξης εφαρμογών Microsoft Visual Studio 2008 [8]

Office, την CardSpace<sup>16</sup>, τη LINQ, μια μεγάλη αναθεώρηση της τεχνολογίας ASP.NET, του CLR (Common Language Runtime, βλ. § 2.3) ενσωματωμένου πλαισίου, και ακόμη περισσότερων. [7]

## 2.6 Το λειτουργικό σύστημα Windows Mobile

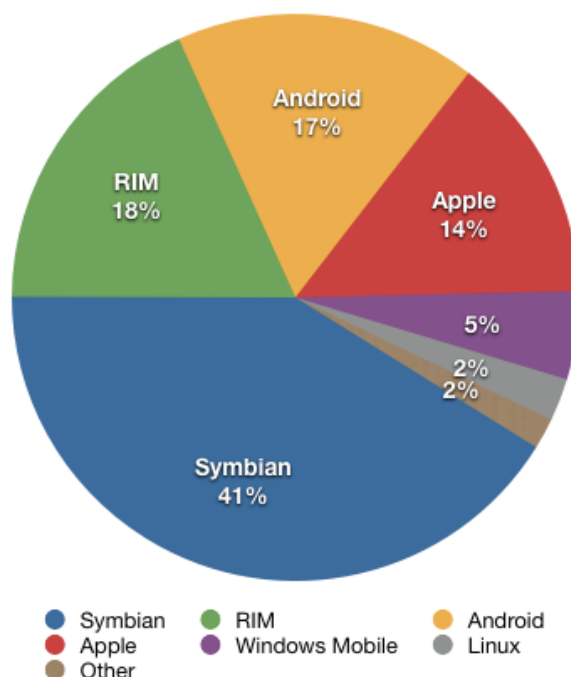
Το Windows Mobile είναι ένα κινητό λειτουργικό σύστημα που αναπτύχθηκε από τη Microsoft για να χρησιμοποιείται στα Smartphones (βλ. § 2.7) και τις κινητές συσκευές (mobile devices, βλ. § 2.7), αλλά καταργείται σταδιακά στις εξειδικευμένες αγορές.

Αρχικά εμφανίστηκε ως λειτουργικό σύστημα με το όνομα Pocket PC 2000 κι οι περισσότερες συσκευές του Windows Mobile ήρθαν με μια πένα τύπου stylus, η οποία χρησιμοποιείται για να εισαγάγει εντολές με το πάτημα σε Touch Screen (Οθόνη Αφής). Η

<sup>16</sup> **Windows CardSpace (με κωδικό όνομα InfoCard):** Είναι ένα λογισμικό πελατών της Microsoft για την ταυτότητα Metasystem (Identity Metasystem). Το CardSpace είναι μια περίπτωση μιας κατηγορίας λογισμικού πελατών ταυτότητας αποκαλούμενου επιλογέα ταυτότητας. Το CardSpace αποθηκεύει τις αναφορές στις ψηφιακές ταυτότητες των χρηστών για τους ίδιους, παρουσιάζοντας τους στους χρήστες ως οπτικές κάρτες πληροφοριών. Το CardSpace παρέχει ένα συνεπές UI (User Interface (= Περιβάλλον Διεπαφής με το Χρήστη) που επιτρέπει στους ανθρώπους να χρησιμοποιήσουν εύκολα αυτές τις ταυτότητες στις εφαρμογές και τους ιστοχώρους όπου γίνονται αποδεκτοί. [21]

Microsoft πρόσφατα ανήγγειλε μια απολύτως νέα τηλεφωνική πλατφόρμα, τη Windows Mobile 7, στο παγκόσμιο συνέδριο κινητών συσκευών στη Βαρκελώνη στις 15 Φεβρουαρίου 2010. Τα τηλέφωνα που έχουν εγκατεστημένη την τρέχουσα έκδοση Windows Mobile 6.x δε θα βελτιώνονται (δε θα έχουν τη δυνατότητα πραγματοποίησης upgrade) στην έκδοση 7.

Το μερίδιο του Windows Mobile της αγοράς των Smartphones έχει πέσει ετησίως, καθώς μειώθηκε κατά 20% στο 3<sup>ο</sup> τρίμηνο του 2009. Πλέον είναι το 5<sup>ο</sup> δημοφιλέστερο λειτουργικό σύστημα για τα Smartphones, με ένα μερίδιο 5% της παγκόσμιας αγοράς (μετά από το Symbian, το BlackBerry OS, το Android και το iPhone). Στις Ηνωμένες Πολιτείες, είναι το 3<sup>ο</sup> δημοφιλέστερο λειτουργικό σύστημα για τα Smartphones για επιχειρησιακή χρήση (μετά από το BlackBerry OS και το iPhone), με ένα μερίδιο 24% μεταξύ των επιχειρηματικών χρηστών.



**Σχήμα 2-4:** Το μερίδιο πωλήσεων του Β' εξαμήνου του 2010 σε τελικούς χρήστες ανά λειτουργικό σύστημα Mobile Device σύμφωνα με το Gartner <sup>[14]</sup>

Η Microsoft καταργεί σταδιακά το Windows Mobile στις εξειδικευμένες αγορές, όπως οι Rugged Devices<sup>17</sup> και εστιάζει στη νέα κινητή πλατφόρμα της, τη Windows Mobile 7.

Υπάρχουν τρεις εκδόσεις του Windows Mobile για τις διάφορες συσκευές υλικού:

- Η Windows Mobile Professional που «τρέχει» σε Smartphones με οθόνη αφής,
- Η Windows Mobile Standard που «τρέχει» σε συσκευές τις κανονικές (τυπικές) οθόνες και
- Η Windows Mobile Classic που «τρέχει» στις κλασικές συσκευές του Windows Mobile, τις Pocket PC. <sup>[21]</sup>

<sup>17</sup> **Rugged Devices (= Τραχείς Συσκευές):** Οι τραχείς συσκευές είναι αυτές οι οποίες αναπτύχθηκαν με σκοπό να λειτουργήσουν ορθά σε σκληρά περιβάλλοντα χρήσης και ρυθμίσεων των συσκευών, όπως σε ισχυρές δονήσεις, σε ακραίες θερμοκρασίες και σε υγρές ή σκονισμένες συνθήκες. <sup>[21]</sup>



## 2.6.1 Η έκδοση Windows Mobile 6.5

Η τρέχουσα έκδοση του λειτουργικού συστήματος Windows Mobile ονομάζεται «Windows Mobile 6.5». Είναι βασισμένο στο Windows CE 5.2<sup>18</sup> Kernel<sup>19</sup>, και χαρακτηρίζεται μια ακολουθία βασικών εφαρμογών που αναπτύσσονται με τη χρήση του Microsoft Windows API<sup>20</sup>. Για τη συγκεκριμένη έκδοση του Windows Mobile, υπήρχε ο σκοπός το λειτουργικό της να είναι κάπως παρόμοιο με τις εκδόσεις desktop των Windows, όσον αφορά τα γνωρίσματά, τη λειτουργία και την αισθητική του. Επιπλέον, η ανάπτυξη λογισμικού από τρίτους (αρχάριους προγραμματιστές) είναι δυνατή και μπορεί να διατεθεί για το Windows Mobile, και οι εφαρμογές αυτές μπορούν να αγοραστούν μέσω του Windows Marketplace για κινητές συσκευές (Windows Marketplace for Mobile<sup>21</sup>).

Το Windows Mobile 6.5 δεν ήταν ποτέ μέρος του σχεδίου ανάπτυξης λειτουργικού για κινητά τηλέφωνα της Microsoft και έχει περιγραφεί από τον κύριο ανώτερο υπεύθυνό του, Steve Ballmer, ως μία «μη πλήρης κι επιθυμητή έκδοση (της Microsoft)», έως ότου η εταιρεία επανέλθει με την τεχνολογία της πολυ-αφής (λειτουργία των touchscreens που μπορεί να αναγνωρίσει παραπάνω από ένα σημεία αφής για την είσοδο εντολών στο κινητό) και την έκδοση του Windows Mobile 7 (που αντικαθίσταται τώρα με το Windows Phone7) και θα εμφανιστεί μάλλον μέσα στο 2010 ή στις αρχές του 2011. Η έκδοση 6.5 είναι μια βελτίωση στο λειτουργικό Windows Mobile 6.1 που εμφανίστηκε για τους κατασκευαστές (εταιρείες κινητών τηλεφώνων) στις 11 Μαΐου 2009, και οι πρώτες συσκευές «έτρεξαν» το λειτουργικό αυτό σύστημα στα τέλη Οκτωβρίου '09.

Αυτή η προσαύξηση ή αναπροσαρμογή περιλαμβάνει μερικά σημαντικά νέα προστιθέμενα χαρακτηριστικά γνωρίσματα, όπως ένα βελτιωμένο GUI (Graphical User Interface = Γραφική Επιφάνεια Διεπαφής), νέα οθόνη με τις κάθετες scrollable ετικέτες (αποκαλούμενες ως «Titanium» (=τιτάνιο)), εν τούτοις θεωρείται γενικά ως δευτερευούσης σημασίας βελτίωση. Περιλαμβάνει επίσης τη νέα κινητή μηχανή αναζήτησης νούμερο 6 του Internet Explorer (browser (=φυλλομετρητής) των Windows), η οποία έχει μια βελτιωμένη διεπαφή σε σχέση με τις προηγούμενες εκδόσεις.

Η Microsoft παρουσίασε αυτήν την έκδοση στο παγκόσμιο συνέδριο κινητών συσκευών του 2009 το Φεβρουάριο και διάφορες συσκευές «τρέχουν» σήμερα αυτήν την έκδοση

---

<sup>18</sup> **Windows CE 5.0:** Τα Windows CE (επίσης γνωστά με την επίσημη ονομασία Windows Embedded Compact (= Windows με Ψηφιακή Ενσωμάτωση) ή Windows Embedded CE (= Ενσωματωμένα Windows CE) και μερικές φορές ως συντετμημένα WinCE) είναι ένα λειτουργικό σύστημα που αναπτύχθηκε από τη Microsoft για τους mini υπολογιστές και τα ενσωματωμένα συστήματα. Τα Windows CE είναι περισσότερο ένα ευδιάκριτο σε μικρές οθόνες λειτουργικό σύστημα και ένα συστατικό πυρήνα (Kernel, βλ. υποσημείωση 19), παρά μια οργανωμένη από τη βάση έκδοση των desktop Windows. Στην έκδοση 5.x, με την κωδική ονομασία «Macallan», που εμφανίστηκε τον Αύγουστο του 2004, προστέθηκαν πολλά καινούργια χαρακτηριστικά όπως: η αυτόματη αποστολή έκθεσης των σφαλμάτων του συστήματος στον κατασκευαστή, Γραφικά τύπου Direct3D για τα κινητά, DirectDraw για τη 2D γραφικά και DirectShow για τη υποστήριξη ψηφιοποίησης ύστερα από λήψη φωτογραφίας ή video, υποστήριξη Remote Desktop Protocol (= Πρωτόκολλο σύνδεσης με Απομακρυσμένο Υπολογιστή Γραφείου (RDP), που αφορά την παροχή ενός χρήστη της γραφικής διεπαφής από έναν άλλο απομακρυσμένο υπολογιστή) κλπ. <sup>[21]</sup>

<sup>19</sup> **Kernel Computing (= Πληροφορική Πυρήνα):** Στην πληροφορική, το Kernel είναι το κεντρικό συστατικό των περισσότερων λειτουργικών συστημάτων υπολογιστών. Είναι μια γέφυρα μεταξύ των εφαρμογών και της επεξεργασίας πραγματικών δεδομένων που γίνεται στο επίπεδο υλικού (= hardware level). <sup>[21]</sup>

<sup>20</sup> **Windows API:** Τα Windows API, ανεπίσημα WinAPI, είναι ένα σύνολο πυρήνων εφαρμογών διεπαφών προγραμματισμού (Application Programming Interfaces) (APIs) της Microsoft διαθέσιμων στα λειτουργικά συστήματα των Microsoft Windows. <sup>[21]</sup>

<sup>21</sup> **Windows Marketplace for Mobile (= Αγορά των Windows για κινητά):** Η αγορά των Windows για κινητά είναι μία υπηρεσία από τη Microsoft για την πλατφόρμα του Windows Mobile που επιτρέπει στους χρήστες για να αναζητήσουν και να μεταφορτώσουν (= κατεβάσουν) εφαρμογές που έχουν αναπτυχθεί από τρίτους. Οι εφαρμογές αυτές είναι διαθέσιμες για χρήση άμεσα στις συσκευές με Windows Mobile 6.5 και στους προσωπικούς Η/Υ. <sup>[21]</sup>

του Windows Mobile, κι είναι αυτές οι συσκευές που «τρέχουν» το Windows Mobile 6.5 και μετά πωλούνται ως «Windows Phones». Μαζί με το Windows Mobile 6.5, η Microsoft ανήγγειλε διάφορες Cloud computing<sup>22</sup> υπηρεσίες με τον κωδικό «SkyBox», «SkyName» ή και «SkyMarket».



**Εικόνα 2-2:** Screenshot του Windows Mobile 6.5 Today Screen "Titanium" <sup>[21]</sup>

Το «SkyBox» έχει επιβεβαιωθεί με το όνομα «My Phone» ενώ το «SkyMarket» με το «Windows Marketplace for Mobile». Μερικές πτυχές της διεπαφής με το χρήστη έχουν ξανασχεδιαστεί, με την αρχική οθόνη (home screen) να μοιάζει με αυτή του φορέα Zune<sup>23</sup> της Microsoft και τη διεπαφή sliding panel (= περιβάλλον ολίσθησης) σύμφωνα με τα πρότυπα της έκδοσης Windows Mobile 6.1. Αυτή η έκδοση σχεδιάστηκε κυρίως για την ευκολότερη χρήση με τα δάχτυλα (δλδ. δυνατότητα αντικατάστασης της χρήσης του stylus pen). Ενώ η συγκεκριμένη έκδοση του Windows Mobile δεν υποστηρίζει εσωτερικά τις χωρητικές οθόνες (capacitive screens), οι κατασκευαστές κινητών τηλεφώνων ήταν σε θέση να την εφαρμόσουν επιτυχώς στις συσκευές τους

Διάφορα τηλέφωνα που τρέχουν αυτήν την περίοδο το Windows Mobile 6.1 είναι ενημερωμένα- βελτιωμένα με την έκδοση στο Windows Mobile 6.5. <sup>[21]</sup>

## 2.7 Mobile Device – Smartphone

Το Mobile Device (Κινητή Συσκευή, επίσης γνωστή ως Handheld Device (= φορητή συσκευή), Handheld Computer (= φορητός υπολογιστής) ή απλά ως Handheld (= φορητός) είναι μια pocket-sized συσκευή υπολογισμού (συσκευή σε μέγεθος τσέπης) που έχει ως

<sup>22</sup> **Cloud Computing:** Το Cloud Computing είναι μέθοδος πληροφορικής βασισμένη στο Διαδίκτυο, σύμφωνα με την οποία οι κοινοί πόροι, το λογισμικό και οι πληροφορίες παρέχονται στους υπολογιστές και σε άλλες συσκευές κατόπιν παραγγελίας, περίπου όπως λειτουργεί και το πλέγμα ηλεκτρικής ενέργειας. <sup>[21]</sup>

<sup>23</sup> **Zune:** Το Zune είναι μια πλατφόρμα ψυχαγωγίας και μια φορητή συσκευή αναπαραγωγής πολυμέσων που αναπτύχθηκε από τη Microsoft. <sup>[21]</sup>

χαρακτηριστικά μια οθόνη επίδειξης με την εισαγωγή εντολής αφής (= touch input) ή/και ένα μικροσκοπικό πληκτρολόγιο. Στην περίπτωση του προσωπικού ψηφιακού βοηθού (= Personal Digital Assistant, PDA) η είσοδος κι η έξοδος εντολών συνδυάζονται συχνά σε μια διεπαφή οθονών επαφής.

Τα Smartphones και τα PDAs είναι δημοφιλή μεταξύ εκείνων που απαιτούν τη βοήθεια και την ευκολία ορισμένων πτυχών ενός συμβατικού υπολογιστή, σε περιβάλλοντα όπου η μεταφορά ενός τέτοιου δε θα καθίστατο πρακτική. Οι επιχειρηματικοί ψηφιακοί βοηθοί (= Enterprise Digital Assistants) μπορούν περαιτέρω να επεκτείνουν τη διαθέσιμη λειτουργία τους, καθώς ένας επιχειρηματίας χρήστης με την προσφέρουν μέσω της κινητής συσκευής τη σύλληψη ολοκληρωμένων δεδομένων τους αναγνώστες γραμμωτών κωδίκων (barcodes), τα RFID<sup>24</sup> και τις έξυπνες κάρτες (= smart cards, κάρτες με ενσωματωμένα ολοκληρωμένα κυκλώματα).<sup>[21]</sup>

Το Smartphone είναι ένα κινητό τηλέφωνο – τύπος μιας Mobile Device που προσφέρει περισσότερο προηγμένες υπολογιστικές δυνατότητες και καλύτερη συνδεσιμότητα σε σχέση με ένα σύγχρονο βασικό τηλέφωνο χαρακτηριστικών γνωρισμάτων (feature phone). Τα Smartphones και τα τηλέφωνα χαρακτηριστικών γνωρισμάτων μπορούν να θεωρηθούν ως φορητοί υπολογιστές που ενσωματώνονται μέσα σε ένα κινητό τηλέφωνο, αλλά ενώ τα περισσότερα τηλέφωνα χαρακτηριστικών γνωρισμάτων είναι σε θέση να «τρέξουν» τις εφαρμογές που είναι βασισμένες σε πλατφόρμες όπως η Java ME, ένα smartphone επιτρέπει στο χρήστη να εγκαταστήσει και να «τρέξει» περισσότερο προηγμένες εφαρμογές βασισμένες σε μια συγκεκριμένη πλατφόρμα. Τα Smartphones «τρέχουν» ένα ολοκληρωμένο λογισμικό λειτουργικού συστήματος που παρέχει μια πλατφόρμα για τους υπεύθυνους για την ανάπτυξη εφαρμογών.



*Εικόνα 2-3: Παράδειγμα Smartphone*

<sup>24</sup> **Radio-frequency identification (= Προσδιορισμός Ραδιοσυχνότητας, RFID):** Είναι η χρήση ενός αντικειμένου (χαρακτηριστικά καλούμενου ως ετικέτα RFID) που εφαρμόζεται ή που ενσωματώνεται σε ένα προϊόν, ένα ζώο, ή ένα πρόσωπο με σκοπό τον προσδιορισμό και τη καταδίωξη χρησιμοποιώντας τα ραδιοκύματα. Μερικές ετικέτες μπορούν να διαβαστούν από διάφορους μετρητές μακριά και πέρα από τη γραμμή θέας του αναγνώστη.<sup>[21]</sup>



Η αύξηση σε ζήτηση για τις προηγμένες κινητές συσκευές που των οποίων οι κατασκευαστές καυχώνται για τους ισχυρούς επεξεργαστές τους, την άφθονη μνήμη, τις μεγαλύτερες οθόνες, και τα ανοικτά λειτουργικά συστήματα έχει ξεπεράσει το υπόλοιπο της κινητής τηλεφωνικής αγοράς για αρκετά έτη. Σύμφωνα με μια μελέτη από την εταιρεία ComScore, πάνω από 45.5 εκατομμύρια άνθρωποι στις Ηνωμένες Πολιτείες είχαν στην κυριότητά τους Smartphones το 2010, κάτι που σημαίνει ότι μιλάμε για έναν ταχύτατης ανάπτυξης τομέα της κινητής τηλεφωνικής αγοράς, η οποία περιέλαβε 234 εκατομμύρια συνδρομητές στις Ηνωμένες Πολιτείες. <sup>[21]</sup>

## 2.8 Ανάπτυξη λογισμικού για το Windows Mobile

Η ανάπτυξη λογισμικού από τρίτους δυνατή για το λειτουργικό σύστημα του Windows Mobile. Υπάρχουν διάφορες επιλογές για τους υπεύθυνους για το σχεδιασμό εφαρμογών που μπορούν να χρησιμοποιήσουν κατά τη διαδικασία της ανάπτυξης μιας εφαρμογής για τα Windows Mobile. Αυτές περιλαμβάνουν το γράψιμο πηγαίου κώδικα με τη γλώσσα C++, το γράψιμο του κώδικα διαχείρισης της εφαρμογής εντός του πλαισίου ανάπτυξης εφαρμογών .NET Compact Framework, ή τη δημιουργία κώδικα που «τρέχει» από την πλευρά του server (server-side code) το αποτέλεσμα του οποίου μπορεί να εμφανιστεί στον Internet Explorer για Mobile Devices ή έναν κινητό πελάτη (mobile client) στη συσκευή του χρήστη. Το .NET Compact Framework είναι στην πραγματικότητα ένα υποσύνολο του .NET πλαισίου και ως εκ τούτου μοιράζεται πολλά συστατικά με την ανάπτυξη λογισμικού σε desktop Η/Υ, τους διακομιστές εφαρμογών και τους κεντρικούς υπολογιστές δικτύου που εγκαθιστά το .NET πλαίσιο, κατά συνέπεια και την ενσωμάτωση του δικτυωμένου υπολογιστικού διαστήματος (δλδ το «Cloud», βλ. υποσημείωση 22). <sup>[21]</sup>

Η ανάπτυξη λογισμικού- εφαρμογών για το Windows Mobile, προϋποθέτει τη χρήση των παρακάτω εργαλείων:

- **Το πρόγραμμα Visual Studio 2005 ή 2008 (πλέον και το 2010):** Το συγκεκριμένο πρόγραμμα χρειάζεται για την ανάπτυξη των εφαρμογών για τα Windows Mobile. Επιτρέπει τη σύνταξη, την αποσφαλμάτωση (debugging) και την αποθήκευση της εφαρμογής για χρήση ή πιθανή παράδοση στο Windows Marketplace for Mobile μέσω ενός εύχρηστου περιβάλλοντος διεπαφής με το χρήστη.
- **Το κατέβασμα (download) του τελευταίου Windows Mobile SDK (Software Development Kit = Εξάρτημα Ανάπτυξης Λογισμικού):** Το SDK περιέχει τα αρχεία επιγραφών και βιβλιοθηκών τύπου API (API header and library files) στα οποία θα πρέπει να έχει κάποιος πρόσβαση για τη λειτουργία του Windows Mobile καθώς επίσης και την τεκμηρίωση, τα προγράμματα αίτησης δειγμάτων, και τους εξομοιωτές που επιτρέπουν την επέκταση και τη διόρθωση της εφαρμογής, ακόμα κι αν ο δημιουργός της δεν έχει μια συσκευή Windows Mobile.
- **Το εργαλείο ActiveSync ή το Windows Mobile Device Center:** Τα συγκεκριμένα εργαλεία είναι απαραίτητα για να εγκατασταθούν και να «παίξουν» οι εφαρμογές στο Mobile Device ή σε έναν εξομοιωτή του.

Για την κατασκευή της εφαρμογής, προϋποτίθεται η δημιουργία ενός Smart Device Project (δλδ. το σχέδιο για την ανάπτυξη εφαρμογής για το Windows Mobile στο πρόγραμμα Visual Studio), η προσθήκη λειτουργικότητας και διαδραστικότητας και το χτίσιμο, το «τρέξιμο» κι η επέκταση της εφαρμογής (μέσω των εργαλείων που διατίθενται από το Visual Studio). <sup>[16]</sup>

## 2.9 Ανάπτυξη εφαρμογών πολιτιστικού πολυμεσικού περιεχομένου με δυνατότητες προσβασιμότητας για Windows Mobile

Η τάση που θέλει σήμερα τους υπολογιστές να μικραίνουν σε μέγεθος, ενώ ταυτόχρονα η δυνατότητά τους για την επεξεργασία δεδομένων να αυξάνεται, έρχεται να δέσει με την τεχνολογία των κινητών τηλεφώνων- συσκευών. Το μεγαλύτερο μέρος του πληθυσμού του ανεπτυγμένου κόσμου στις μέρες μας είναι συνδρομητής μιας εταιρείας παροχής υπηρεσιών κινητής τηλεφωνίας, ή τουλάχιστον κάτοχος μιας Κινητής Συσκευής (Mobile Device). Το δεδομένο αυτό ήρθαν να εκμεταλλευτούν οι μεγάλες εταιρείες του χώρου της πληροφορικής (Google, Apple, Microsoft κλπ.) για να προσαρμόσουν ή και να επεκτείνουν τις υπηρεσίες του προς το συγκεκριμένο τομέα- τεχνολογία. Σήμερα, αλλά πολύ περισσότερο ακόμα στα αμέσως επόμενα χρόνια, η τεχνολογία αυτή θα επικρατήσει και θα καταλάβει ακόμη μεγαλύτερο μερίδιο της αγοράς, κάτι που μεταφράζεται σε εμπορική άρα και σε οικονομική επιτυχία.

Όλα τα παραπάνω εργαλεία που ως ένα βαθμό αναλύθηκαν, αποτελούν ένα πολύ ισχυρό σύνολο στα χέρια των σχεδιαστών λογισμικού και συστημάτων για την κατασκευή εφαρμογών. Η ανάπτυξη εφαρμογών για το λειτουργικό Windows Mobile της Microsoft είναι μία δυνατότητα από τις πολλές των συγκεκριμένων εργαλείων, σχετικά νέα, αλλά συνεχώς και ραγδαίως εξελισσόμενη.

Είναι γεγονός ότι ο τομέας του πολιτισμού μπορεί να εκμεταλλευτεί (όπως και γίνεται ως ένα βαθμό) τη δυνατότητα αυτή, ώστε να παράσχει νέες υπηρεσίες και διευκολύνσεις που θα αξιοποιούν τους υπολογιστές χειρός, και ταυτόχρονα κινητά τηλέφωνα, για να παρουσιάσει τα διάφορα θέματά του. Κάτι τέτοιο θα σήμαινε αυτόματα την μεγαλύτερη προώθηση των πολιτιστικών προϊόντων του και την ενημέρωση και πληροφόρηση του κόσμου γύρω από το περιεχόμενο που θέλει να παρουσιάσει με ένα νέο, ίσως πιο ευχάριστο τρόπο. Είναι για παράδειγμα υπόθεση ρουτίνας στις μέρες μας κάποιος, ακόμα και μη έμπειρος χρήστης, να φτιάξει μια εφαρμογή Windows Mobile με πολιτιστικό περιεχόμενο που να ενσωματώνει και πολυμέσα και να την «ανεβάσει» στο Windows Marketplace for Mobile, όπου μπορεί να την πουλήσει ή να την διανείμει δωρεάν, πετυχαίνοντας έτσι και πιθανή είσπραξη χρημάτων, αλλά ακόμη περισσότερο προβολή του πολιτιστικού περιεχομένου στο οποίο αναφέρεται η εφαρμογή. Όπως είναι επίσης εύκολο, ένα μουσείο να δημιουργήσει πάλι μια τέτοια εφαρμογή, ώστε οι επισκέπτες του να έχουν τη δυνατότητα να την εγκαταστήσουν στις Mobile Devices τους που ούτως ή άλλως κάποιοι θα έχουν και να απολαμβάνουν με αυτό τον τρόπο μια εναλλακτική περιήγηση στο μουσείο ή να δέχονται πρόσθετες πληροφορίες γύρω από τα εκθέματά του.

Σύμφωνα όμως και με τα δεδομένα που αναλύθηκαν και αναφέρονται στα άτομα με αναπηρίες, ένας σχεδιαστής λογισμικού οφείλει να πειθαρχεί και με τουλάχιστον κάποιους βασικούς κανόνες προσβασιμότητας. Αν και η προσπάθεια ανάπτυξης λογισμικού με προσβάσιμο περιεχόμενο δυσκολεύει στην περίπτωση που αυτό προορίζεται για Mobile Devices, υπάρχει εντούτοις η δυνατότητα και με τη βοήθεια των παραπάνω εργαλείων το συγκεκριμένο εγχείρημα να πραγματοποιηθεί.

Αυτές ακριβώς τις δυνατότητες έρχεται να προτείνει με ένα παράδειγμα ανάπτυξης τέτοιου λογισμικού η εφαρμογή που παρουσιάζεται παρακάτω.

# Κεφάλαιο 3: Ανάπτυξη και Παρουσίαση της Εφαρμογής

---

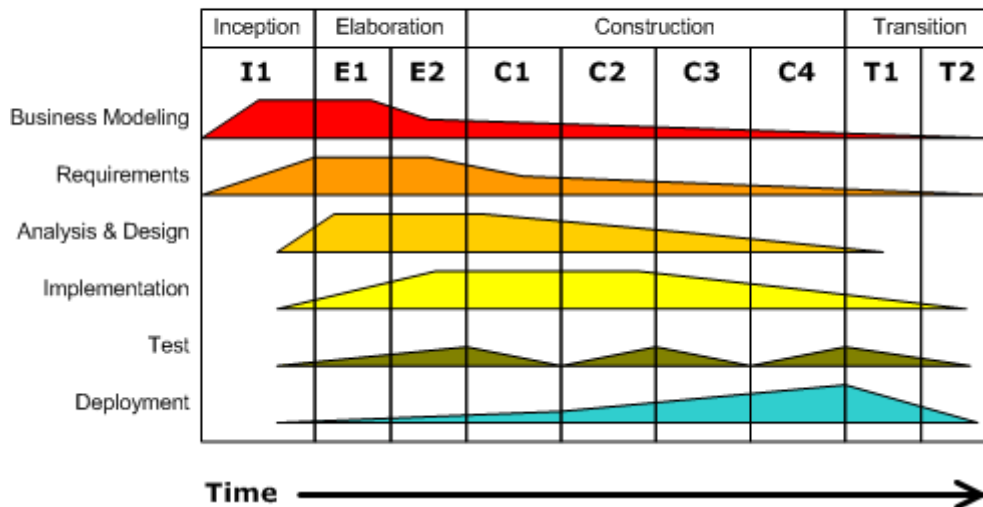
## 3.1 Μοντέλο διαδικασίας λογισμικού που χρησιμοποιήθηκε

Ως κλάδος της πληροφορικής, η τεχνολογία λογισμικού διαθέτει διάφορα μοντέλα διαδικασίας ανάπτυξης. Το μοντέλο διαδικασίας λογισμικού (software process model) είναι μια απλοποιημένη περιγραφή μιας διαδικασίας ανάπτυξης λογισμικού που παρουσιάζεται υπό ορισμένη οπτική γωνία. Το μοντέλο της διαδικασίας του λογισμικού περιγράφει τον τόπο οργάνωσης των δραστηριοτήτων ανάπτυξης (απαιτήσεις, ανάλυση, σχεδίαση, κωδικοποίηση, έλεγχος) του λογισμικού. Τα μοντέλα εκ φύσεως είναι απλοποιήσεις, και άρα ένα μοντέλο διαδικασίας λογισμικού είναι μια αφηρημένη αναπαράσταση μίας προσέγγισης ανάπτυξης. Εφαρμόζεται προσεγγιστικά και ανάλογα με την κατηγορία του λογισμικού. <sup>[12]</sup>

Για την παρούσα ανάπτυξη της εφαρμογής για τη διερεύνηση δυνατοτήτων προσβασιμότητας στο λειτουργικό σύστημα Windows Mobile, χρησιμοποιήθηκε το **επαναληπτικό μοντέλο** διαδικασίας λογισμικού. Σύμφωνα με αυτό, η ανάπτυξη του λογισμικού οργανώνεται σε **επαναλήψεις** (iterations), όπου σε κάθε επανάληψη εκτελούνται όλες οι δραστηριότητες της ανάπτυξης, ξεκινώντας από τον προσδιορισμό των απαιτήσεων και καταλήγοντας στην κωδικοποίηση και στον έλεγχο. Κάθε επανάληψη μπορεί να θεωρηθεί ως μία μικρογραφία ενός έργου ανάπτυξης που οδηγεί στην παραγωγή ημιτελών εκτελέσιμων προγραμμάτων, τα οποία όμως έχουν ελεγχθεί με επιτυχία. Το επαναληπτικό μοντέλο είναι εξ ορισμού και επαυξητικό (incremental), επειδή η λειτουργικότητα προστίθεται σταδιακά στο λογισμικό. Όπως φαίνεται και στο σχήμα 3-1, σε κάθε επανάληψη πραγματοποιούνται όλες οι δραστηριότητες αλλά με διαφορετική βαρύτητα. Στις πρώτες επαναλήψεις η προσοχή εστιάζεται στον προσδιορισμό των απαιτήσεων και στην αρχιτεκτονική σχεδίαση. Όσο προχωρά η ανάπτυξη, το βάρος μετατοπίζεται στη λεπτομερή σχεδίαση, στην κωδικοποίηση και στον έλεγχο. <sup>[12]</sup>

Τα προγράμματα που παράγονται σε κάθε επανάληψη αποτελούν πρωτότυπα τα οποία χρησιμοποιούνται ως μέσο ανατροφοδότησης. Τα πρωτότυπα αυτά επαυξάνονται με πρόσθετη λειτουργικότητα σε κάθε επανάληψη και καταλήγουν στην τελική έκδοση του λογισμικού. Η διαδικασία αυτή πολλές φορές καλείται και **εξελικτική πρωτοτυποποίηση** (evolutionary prototyping). <sup>[12]</sup>

**Iterative Development**  
Business value is delivered incrementally in time-boxed cross-discipline iterations.



Σχήμα 3-1: Επαναληπτικό μοντέλο ανάπτυξης λογισμικού <sup>[21]</sup>

### 3.1.1 Γιατί επαναληπτικό μοντέλο για την εφαρμογή;

Οι λόγοι για τους οποίους επιλέχθηκε το επαναληπτικό μοντέλο για την ανάπτυξη της εφαρμογής, είναι οι παρακάτω:

- Το επαναληπτικό μοντέλο είναι κατάλληλο για προβλήματα στα οποία οι απαιτήσεις δεν είναι γνωστές ή ενδέχεται να αλλάζουν. <sup>[12]</sup> Στην παρούσα εφαρμογή, αν κι ήταν σε γενικές γραμμές γνωστές οι απαιτήσεις των χρηστών μέσα στους οποίους εντάσσονται και άτομα με αναπηρίες, δεν ήταν εξακριβωμένο (τουλάχιστον από την αρχή), ποιων κατηγοριών ατόμων με αναπηρίες θα ικανοποιούσε την ανάγκη για πρόσβαση, πως θα διευκολύνονταν και θα είχαν μια ευχάριστη πλοήγηση άτομα χωρίς αναπηρίες και πως ακριβώς οι δύο παραπάνω στόχοι θα ικανοποιούταν παράλληλα.
- Ένα βασικό πλεονέκτημα του επαναληπτικού μοντέλου, είναι ότι παρέχει θετική αντιμετώπιση των κινδύνων του έργου. Αντιμετωπίζονται άριστα οι κίνδυνοι των απαιτήσεων και αρκετά καλά οι κίνδυνοι που αφορούν την αρχιτεκτονική. Επίσης αντιμετωπίζονται πολύ καλά οι κίνδυνοι που αφορούν την ποιότητα του τελικού προϊόντος, επειδή οι έλεγχοι πραγματοποιούνται σε κάθε επανάληψη κι όχι στην τελική φάση της κωδικοποίησης. <sup>[12]</sup> Σε περιπτώσεις ειδικά, όπως κι η συγκεκριμένη, όπου δεν υπάρχει μεγάλη ή επαγγελματική εμπειρία από την ανάπτυξη αντίστοιχων έργων, το επαναληπτικό μοντέλο κρίνεται ως ιδιαίτερα χρήσιμο.
- Μετά την περάτωση κάθε κύκλου ανάπτυξης της εφαρμογής, είναι βολικό να γίνεται εκ νέου έλεγχος που να περιλαμβάνει κατά πόσο η αυτή έχει ικανοποιήσει ως εκείνο το σημείο τους αρχικούς στόχους που τέθηκαν, ποια είναι η πρόοδος στην ενσωμάτωση δυνατοτήτων προσβασιμότητας, να διευθετηθούν πιθανές βελτιώσεις που χρειάζονται (κάτι που συμβαίνει σχεδόν πάντοτε), να ορισθούν αν κριθεί αναγκαίο νέοι στόχοι.
- Το επαναληπτικό μοντέλο επίσης, ενθαρρύνει την άμεση συμμετοχή των χρηστών σε όλη τη φάση της ανάπτυξης του έργου. Μετά από κάθε ολοκλήρωση ενός

κύκλου, είναι θεμιτή η βοήθειά τους για να εξακριβωθεί ο βαθμός ικανοποίησής τους, να προτείνουν ιδέες βελτίωσης και πρακτικές που θα διευκολύνουν την πλοήγησή τους. Η συμμετοχή των χρηστών αποκτά ακόμη μεγαλύτερη σημασία, όταν το έργο απευθύνεται και ειδικά σε άτομα με αναπηρίες.

### 3.2 Επιλογή θέματος παρουσίασης για την εφαρμογή

Ο βασικός σκοπός της ανάπτυξης της εφαρμογής, ήταν ως γνωστόν η διερεύνηση για τις δυνατότητες προσβασιμότητας που μπορεί να παρέχει το λειτουργικό σύστημα Windows Mobile. Η βάση του συγκεκριμένου στόχου, είναι δυνατόν να διαπιστωθεί καλύτερα, αν ικανοποιηθεί μια ανάγκη από την καθημερινότητα κοινών χρηστών, αλλά και ατόμων με αναπηρίες που θα έχουν την ίδια και ειδικότερα προσαρμοσμένη ανάγκη για πρόσβαση στο προϊόν λογισμικού που παρουσιάζεται.

Η παρουσίαση ενός πολιτιστικού θέματος, κρίθηκε ότι μπορεί να επιτελέσει τον παραπάνω ρόλο. Η ισότιμη πρόσβαση σε στοιχείο και τομείς πολιτισμού, είναι γνωστό ότι αποτελεί ένα κρίσιμο ζήτημα για τον ανθρωπισμό της σημερινής εποχής. Το ίδιο βεβαίως ισχύει και για τα προϊόντα λογισμικού που αναφέρονται σε αυτόν και παρουσιάζουν ένα πολιτιστικό θέμα.

Όπως έχει προαναφερθεί, κάλλιστα ένας οργανισμός ή ένα μουσείο, μπορεί να αναθέσει σε μηχανικούς λογισμικού και συστημάτων πληροφορικής να κατασκευάσουν μια εφαρμογή που θα προτείνει έναν εναλλακτικό, ιδιαίτερο κι ευχάριστο τρόπο πλοήγησης σε μια έκθεση ή σε μια εφαρμογή που θα διαμοιράζεται στους χρήστες μέσω του Windows Marketplace. Και μιας κι η τάση είναι ούτως ή άλλως τα κινητά τηλέφωνα, πέραν του κλασικού ρόλου τους που τα έχουμε συνηθίσει, να γίνονται κι οι υπολογιστές χειρός, η ανάπτυξη λογισμικού προσανατολίζει μεγάλο κομμάτι της έρευνας και της δημιουργίας της προς τα εκεί.

Η εκμετάλλευση μάλιστα των νέων τεχνολογιών, όπως είναι και τα λειτουργικά άλλωστε για Mobile Devices από φορείς πολιτισμού, μπορούν να εκτινάξουν την προώθηση πολιτιστικών προϊόντων και σε επίπεδο αγοράς, αλλά πολύ περισσότερο σε επίπεδο πολιτιστικής ανέλιξης της κοινωνίας, ιδιαίτερα των νέων ηλικιών.

Έτσι, αποφασίστηκε το θέμα που θα παρουσιαστεί να είναι η **Αρχαία Ελληνική Τραγωδία**, με τον ιδιόμορφο παραπάνω τρόπο. Και να «τρέχει» δηλαδή η εφαρμογή (προορίζεται βασικά) για το λειτουργικό σύστημα Windows Mobile, αλλά και να ενσωματώνονται σ' αυτή δυνατότητες προσβασιμότητας. Το θέμα επίσης επιλέχθηκε διότι είναι μαζεμένο, μπορεί δηλαδή να αποφευχθεί η πληθώρα πληροφοριών που πιθανώς να κουράσουν το χρήστη, χωρίζεται σε σαφή κομμάτια- θέματα που ως ένα βαθμό ακολουθεί κι η ίδια η εφαρμογή, μπορεί εύκολα να ενημερωθεί με νέο υλικό (ούτως ή άλλως τα εργαλεία υλοποίησης βοηθούν σ' αυτό). Προωθεί επίσης στοιχεία πολιτισμού της χώρας μας. Μπορεί εν τέλει να επιτελέσει ως θέμα, όλους τους αρχικούς στόχους που τέθηκαν, με την κατάλληλη βέβαια αξιοποίηση των εργαλείων υλοποίησης. Δηλαδή την ευχάριστη εμπειρία από πλευράς χρήστη, την προσβασιμότητα, το σαφές διαρθρωμένο περιεχόμενο, την ενσωμάτωση πολυμέσων που μπορούν να υποβοηθήσουν μάλιστα την προσβασιμότητα, να θέσει μία πρόταση για την προώθηση πολιτιστικών στοιχείων και θεμάτων του πολιτισμού σε ένα ευρύ κοινό με νέο τρόπο και νέα μέσα.

Όλα αυτά μαζί συντελούν στο στόχο της σχεδίασης για όλους μέσω της ανάπτυξης μιας εφαρμογής με πολιτιστικό περιεχόμενο σε Windows Mobile.

### 3.3 Ανάλυση απαιτήσεων εφαρμογής

Κάθε προϊόν λογισμικού, έτσι και το παρόν, προορίζονται για διάφορους σκοπούς. Παρακάτω, παρατίθενται ανά κατηγορία οι στόχοι που φιλοδοξεί να ικανοποιήσει, και ικανοποιεί τελικά, η εφαρμογή:

#### 3.3.1 Φυσικό περιβάλλον

Το σύστημα που αναπτύχθηκε, προορίζεται για να εγκατασταθεί σε Mobile Devices με εγκατεστημένο το λειτουργικό σύστημα Windows Mobile Professional 6 SDK, καθώς και στις νεότερες από αυτήν εκδόσεις. Στην εφαρμογή θα υπάρχουν τόσες θέσεις πρόσβασης, όσες ο χρήστης το έχει εγκατεστημένο σε διαφορετικό χώρο αποθήκευσης αρχείων στη Mobile Device του. Δεν υπάρχουν συγκεκριμένοι περιβαλλοντικοί περιορισμοί (πχ θερμοκρασία, υγρασία κλπ), πέρα από αυτούς που θέτει ο κατασκευαστής της Mobile Device, εκτός ίσως από την έντονη φωτεινότητα όπου δε θα είναι ευδιάκριτα τα χρώματα στην εφαρμογή. Ο βέλτιστος βέβαια τρόπος αξιοποίησης των δυνατοτήτων της εφαρμογής ιδιαίτερα από άτομα με προβλήματα όρασης, επιτυγχάνεται με την εγκατάσταση ενός screen reader για Mobile Device.

#### 3.3.2 Χρήστες κι εργονομικοί παράγοντες

Το σύστημα πρόκειται να το χρησιμοποιήσουν κοινός χρήστες που διαθέτουν ή μπορούν να έχουν κάποιου είδους πρόσβαση σε Mobile Device με εγκατεστημένο το λειτουργικό Windows Mobile Professional 6+ SDK.

Οι μόνοι ουσιαστικά χρήστες που εξαιρούνται από τη δυνατότητα λειτουργία της εφαρμογής, είναι εκείνοι που καθόλου έως ελάχιστη εξοικείωση με τους Η/Υ ή τα κινητά τηλέφωνα, τα άτομα με ιδιαίτερα νοητικά προβλήματα κι εκείνοι οι οποίοι έχουν σοβαρή παράλυση ή πρόβλημα στα χέρια (με διάφορες βέβαια ειδικές εξαιρέσεις όπως πχ τη χρησιμοποίηση stylus pen με το στόμα).

Ο κοινός χρήστης με εξοικείωση στα σύγχρονα προϊόντα λογισμικού δε θα έχει κανένα πρόβλημα να χρησιμοποιήσει την εφαρμογή. Επίσης άτομα με προβλήματα ακοής, ομιλίας και κίνησης των κάτω άκρων δε θα έχουν να αντιμετωπίσουν κάποιο πρόβλημα. Τα άτομα με προβλήματα όρασης, απλά θα πρέπει να εξοικειωθούν με την ιδέα του screen reader (αν δεν το έχουν κάνει ήδη). Τέλος, οι χρήστες με νοητικά προβλήματα, θα μπορούν να τη χρησιμοποιήσουν στο βαθμό που μπορούν να πλοηγηθούν, και ακόμη φυσικά πιο ολοκληρωμένα, να κατανοήσουν τα κείμενα που περιέχονται στην εφαρμογή.

Δεν απαιτείται κάποια συγκεκριμένη κατάρτιση από τους χρήστες για τη χρήση της. Μόνο, όπως προαναφέρθηκε να έχουν μία έστω ελάχιστη εξοικείωση με Η/Υ ή κινητά τηλέφωνα γενικά (και σε αυτή την περίπτωση η χρήση δεν καθίσταται απαγορευτική).

Η κατανόηση και η χρήση του συστήματος από το χρήστη είναι ιδιαίτερα εύκολη. Επίσης η πιθανότητα κακής χρήσης του είναι μηδενική.



### 3.3.3 Λειτουργικότητα

Η εφαρμογή θα αλλάζει φόρμες και θα παραθέτει πληροφορίες για αυτό που ζητάει ο χρήστης (και εννοείται πως περιλαμβάνεται σε αυτή) όταν και όπως εκείνος το θέλει (από τους τρόπους πλοήγησης που υπάρχουν).

Η τροποποίηση κι η επέκταση του συστήματος είναι δυνατή μόνο μετά από παρέμβαση του δημιουργού και εκ νέου εγκατάστασης της νέας πια εφαρμογής στη Mobile Device. Οποιαδήποτε τροποποίηση και επέκταση δεδομένων και στοιχείων από το χρήστη δεν είναι δυνατή.

Δεν υπάρχουν επίσης περιορισμοί στην ταχύτητα εκτέλεσης ή στο χρόνο απόκρισης του συστήματος.

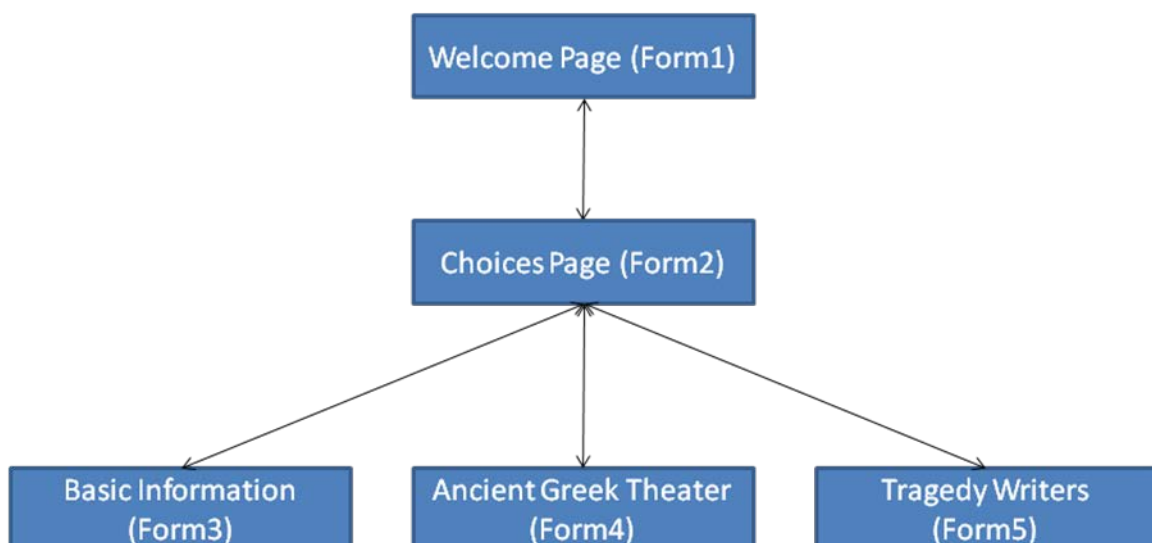
### 3.3.4 Λοιπές απαιτήσεις

Δεν υπάρχουν συγκεκριμένες απαιτήσεις ασφάλειας, πέρα από τη μη παροχή δυνατότητας του χρήστη τροποποίησης των δεδομένων και λειτουργιών της συσκευής. Ακόμη και στα αντικείμενα τύπου input που υπάρχουν όπως τα Text Boxes (χώρος εισόδου/ εξόδου κειμένου) και το Combo Box (λίστα επιλογής), στα μεν πρώτα είναι συγκεκριμένο το κείμενο- δεδομένο που θα εμφανίζεται ανάλογα με τις επιλογές και την πλοήγηση του χρήστη και το δεύτερο είναι συγκεκριμένες τρεις οι δυνατές επιλογές.

Η μετακίνηση της εφαρμογής από τοποθεσία σε τοποθεσία, από μία Mobile Device σε μία άλλη, από Η/Υ σε έναν άλλο ή από Η/Υ σε Mobile Device και τούμπαλιν είναι τυπικής- γνωστής διαδικασίας μεταφοράς δεδομένων και αρχείων.

## 3.4 Πλοήγηση

Η πλοήγηση στην εφαρμογή είναι απλή και συνηθισμένη. Μια απλή εμπειρία του χρήστη με τις νέες τεχνολογίες, είναι αρκετή ως εξοικείωση κι ικανότητα για να πλοηγηθεί. Το ίδιο εύκολο είναι κι η κατανόησή της. Διατηρείται ο ίδιος σχεδόν τρόπος πλοήγησης μέσα στην εφαρμογή από φόρμα σε φόρμα με ελάχιστες εξαιρέσεις. Έτσι ο χρήστης δεν ξεκινάει κάθε φορά από το μηδέν, διευκολύνεται συνολικότερα και δίνει έμφαση στο περιεχόμενο. Η πλοήγηση αναπαριστάται στο παρακάτω διάγραμμα.



### Σχήμα 3-2: Διάγραμμα πλοήγησης εφαρμογής

Μπαίνοντας στην εφαρμογή, ο χρήστης εισάγεται αρχικά στη φόρμα «καλωσορίσματος» Welcome Page. Από εκεί, έχει μόνο μία επιλογή, να εισέλθει στο «κέντρο» της πλοήγησης και στη φόρμα Choices Page. Από εκεί, δίνεται η δυνατότητα εισόδου στις άλλες τρεις φόρμες που περιλαμβάνουν ουσιαστικά και το περιεχόμενο της εφαρμογής, στις Basic Information, Ancient Greek Theater και Tragedy Writers. Σε κάθε φόρμα, πέραν της αρχικής, ο χρήστης μπορεί να επιστρέψει στην προηγούμενη φόρμα όπου βρισκόταν.

#### 3.4.1 Βασική πλοήγηση

Η βασική πλοήγηση, πραγματοποιείται μέσω του Main menu (= βασικό μενού). Η βασική πλοήγηση βασίζεται στη χρήση βασικού μενού (= main menu) που είναι ένα tool (= εργαλείο) για τις Mobile Devices που υπάρχει by default. Τα menu items (= αντικείμενα του μενού) που δημιουργούνται όταν θέλει κάποιος να αξιοποιήσει το main menu (όπως εδώ), λειτουργούν ως κουμπιά, που κάνοντας κλικ σε αυτά, πραγματοποιούν συγκεκριμένες λειτουργίες. Δύο είναι οι κατηγορίες των menu items, της πρώτης και της δεύτερης σειράς, με τη δεύτερη να εμφανίζεται, εφόσον υπάρχει, κάνοντας κλικ στα menu items της πρώτης που έχουν βελάκι προς τα δεξιά.



Εικόνα 3-1: Παράδειγμα Main Menu σε πλήρη αναδίπλωση

Ο χρήστης, κάνοντας κλικ σε κάποιο από αυτά τα menu items, εμφανίζει στη φόρμα τα αντίστοιχα αντικείμενα και τις πληροφορίες που τα τελευταία διαθέτουν. Στην περίπτωση που το menu item της πρώτης σειράς «δείχνει» σε άλλα δεύτερης, το πρώτο επιτελεί ακριβώς αυτή τη λειτουργία, με τα menu items της δεύτερης σειράς να είναι υπεύθυνα τώρα για την εμφάνιση του κατάλληλου περιεχομένου στη φόρμα.

Το main menu χρησιμοποιείται σε όλες τις φόρμες. Σε αυτό περιέχεται επίσης και το menu item back, που είναι υπεύθυνο για την επιστροφή στην προηγούμενη φόρμα. Το ίδιο βέβαια δεν ισχύει για την πρώτη φόρμα, όπου ένα τέτοιο menu item θα ήταν περιττό. Η έξοδος από την εφαρμογή (terminate application) γίνεται με το κουμπί X πάνω δεξιά στη μπάρα κορυφής που διαθέτουν από μόνες τους οι Mobile Devices, γι' αυτό δε χρειάζεται για το συγκεκριμένο σκοπό να γίνει κάποια προσθήκη εργαλείου και λειτουργίας.

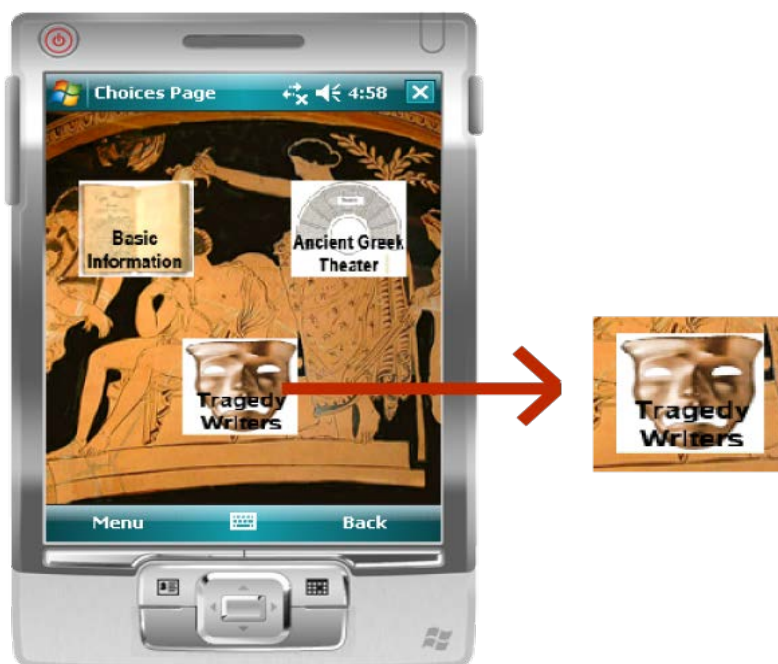


### 3.4.2 Εναλλακτική πλοήγηση

Η εναλλακτική πλοήγηση τελείται με δύο τρόπους:

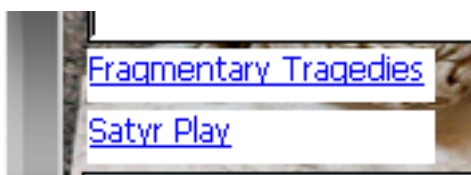
- Με τη χρήση εικόνων που λειτουργούν ως buttons (= κουμπιά) και
- με τη χρήση link label (= ετικετών συνδέσμων)

Η πρώτη περίπτωση συναντάται στη φόρμα Choices Page, όπου η χρήση των εικόνων, παραπέμπει ουσιαστικά στην κλασική λειτουργία του μενού των κινητών τηλεφώνων, επομένως, είναι γνωστή ούτως ή άλλως για τους χρήστες μέθοδος πλοήγησης. Κάθε εικόνα- κουμπί από αυτά, εισάγε το χρήστη στην αντίστοιχη φόρμα, όπως ακριβώς γίνεται και με τα menu items του main menu της φόρμας αυτής. Η εναλλακτική πλοήγηση μάλιστα εδώ, είναι βολικότερη σε σχέση με τη βασική και φαίνεται στην επόμενη εικόνα.



*Εικόνα 3-2: Φόρμα Choices Page με έμφαση στην εναλλακτική πλοήγηση*

Η δεύτερη περίπτωση συναντάται στις φόρμες που εμφανίζουν το περιεχόμενο της εφαρμογής Basic Information, Ancient Greek Theater και Tragedy Writers. Εδώ, τα links (από τα tools του Visual Studio) που χρησιμοποιούνται, παίζουν κι αυτά το ρόλο μιας πιο εύκολης κι άμεσης πλοήγησης κι εναλλαγής περιεχομένου. Το περιεχόμενο που καθορίζουν είναι αυτό του text box (= πλαισίου κειμένου), κάτι που γίνεται αντιληπτό από το text των ίδιων των link labels. Οι σύνδεσμοι εμφανίζονται όταν γίνει κλικ σε ένα menu item δεύτερης σειράς και αποκρύπτονται όταν το περιεχόμενο της φόρμας δε σχετίζεται με κάποιο από αυτά.



*Εικόνα 3-3: Link Labels που χρησιμοποιούνται ως εναλλακτική πλοήγηση*

Η αλήθεια είναι πως στη συγκεκριμένη εφαρμογή, η εναλλακτική πλοήγηση τυχαίνει είναι πιο βολική από τη βασική ή τουλάχιστον με αυτό τον τρόπο αξιοποιείται. Θεωρήθηκε πάντως σκόπιμο κατά την ανάπτυξή της, να αξιοποιηθεί η bottom bar (κάτω μπάρα) της Mobile Device που έτσι κι αλλιώς υπάρχει κι είναι μια κλασική μέθοδος πλοήγησης του Windows Mobile.

### 3.5 Φόρμες εφαρμογής

Η εφαρμογή αποτελείται από πέντε στον αριθμό φόρμες (ή αλλιώς σελίδες). Κάθε φόρμα έχει το δικό της όνομα το οποίο αντιστοιχεί στο περιεχόμενο που αυτή εμφανίζει. Το όνομα της φόρμας, αποτελεί ένα property (= στοιχείο), το οποίο καθορίζεται από το Property Window του Visual Studio.

Καλό είναι να σημειωθεί εδώ, πως όταν αναφερόμαστε στο όνομα της φόρμας, εννοούμε τον τίτλο- κείμενό της που φαίνεται ως όνομα στην κορυφή της σελίδας στη συσκευή. Στην ουσία επομένως μιλάμε για το κείμενο που αναγράφεται στην μπάρα στην κορυφή (top bar) του Mobile Device και καθορίζεται από τα properties-> text του Property Window. Το πραγματικό όνομα της φόρμας που καθορίζεται από τα properties-> name, είναι αυτό που χαρακτηρίζει μονοσήμαντα το αντικείμενο Form (= φόρμας) και με αυτό το όνομα, γίνεται αναγνωρίσιμο στον κώδικα.



**Εικόνα 3-4:** Το top bar του Mobile Device όπου αναγράφεται κι ο τίτλος της παρούσας φόρμας

Οι φόρμες της εφαρμογής κι η λειτουργία που η καθεμιά τους επιτελεί, είναι οι εξής:

- **Form1-> Welcome Page:** Η φόρμα καλωσόρισης που εισάγει μέσω ενός κουμπιού το χρήστη στο θέμα και στη λειτουργία της εφαρμογής.
- **Form2-> Choices Page:** Στη συγκεκριμένη φόρμα, υπάρχει το κεντρικό μενού της εφαρμογής. Μέσω αυτού, ο χρήστης μπορεί να εισαχθεί στις ενότητες που παρουσιάζονται, Basic Information, Ancient Greek Theater και Tragedy Writers.
- **Form3-> Basic Information:** Αυτή η φόρμα, όπως υποδηλώνει άλλωστε και το όνομά της, διαθέτει κάποιες βασικές πληροφορίες για την Αρχαία Ελληνική Τραγωδία και ο χρήστης της εφαρμογής πρέπει να γνωρίζει. Αυτές διακρίνονται στις κατηγορίες Tragedy, Performance of Greek Tragedies και Origin of Tragedy.
- **Form4-> Ancient Greek Theater:** Περιέχει πληροφορίες γύρω από το Αρχαίο Ελληνικό Θέατρο όπου λάμβαναν χώρα οι Τραγωδίες με τις κατηγορίες Ancient Greek Theater και Theater Buildings όπου με τη σειρά του χωρίζεται στις υποκατηγορίες Theater Buildings, Audience, Skene και Orchestra.
- **Form5-> Tragedy Writers:** Αφορά τους γνωστούς εκπροσώπους της Αρχαίας Ελληνικής Τραγωδίας Αισχύλο, Σοφοκλή κι Ευριπίδη. Μέσω μιας αναδιπλούμενης λίστας ο χρήστης επιλέγει έναν από τους τρεις και αυτομάτως παρέχεται η δυνατότητα ενημέρωσης με πληροφορίες γύρω από τις κατηγορίες Commentary, Works και Life για τον κάθε ποιητή.

Για κάθε φόρμα υπάρχει επίσης αντίστοιχο με το όνομα και τη χρήση της tag (= ετικέτα) που ορίζεται από τα properties-> tag του Property Window.

### 3.6 Εξοικείωση του χρήστη με το περιβάλλον της εφαρμογής

Για το περιβάλλον της εφαρμογής, δεν απαιτείται κάποια ιδιαίτερη εξοικείωση, κάτι το οποίο άλλωστε αποτελούσε και στόχο για την επιτυχή σχεδίαση. Εκείνο για το οποίο θα πρέπει να μεριμνά πάντα ο χρήστης είναι να ξέρει ανά πάσα στιγμή που βρίσκεται. Γι' αυτό βέβαια, υποβοηθείται από την ίδια τη σχεδίαση του λογισμικού. Για να κατατοπίζεται λοιπόν σωστά και να συλλέγει την πληροφορία που τον ενδιαφέρει, πρέπει να έχει στα υπ' όψιν του τα εξής πράγματα:

- Το όνομα της φόρμας που «δείχνει» σε ποια είναι και τι μπορεί κανείς να περιμένει να δει μέσα σε αυτή (το θέμα της- σκοπό της).
- Τους τίτλους- επικεφαλίδες κάθε text box (πλαισίου κειμένου) ή εικόνας (βλ. § 3.8).
- Το κεντρικού μενού (main menu) που βρίσκεται κάτω κάτω στην κάθε φόρμα και μέσω αυτού γίνεται η βασική πλοήγηση (βλ. § 3.4.1).
- Την εναλλακτική πλοήγηση που υπάρχει (βλ. § 3.4.2).

Με τη σωστή χρήση αυτών όλων των αντικειμένων, είναι δεδομένο πως ο χρήστης δε θα συναντήσει κάποιο πρόβλημα. Το μόνο βεβαίως που τίθεται βεβαίως εδώ, είναι για το ποια από τις δύο πλοηγήσεις που βρίσκεται στην κάθε φόρμα τον βολεύει, αν και γενικότερα, όπως αναφέρθηκε και προηγουμένως, η εναλλακτική σχεδιάστηκε για να είναι πιο πρακτική.

### 3.7 Ενσωμάτωση αρχών προσβασιμότητας

Οι τακτικές προσβασιμότητας, υιοθετήθηκαν από το πρόγραμμα του World Wide Web Consortium (W3C), Web Accessibility Initiative και τα χαρακτηριστικά- προϋποθέσεις προσβασιμότητας που αυτό έχει καθορίσει μέσω του Web Content Accessibility Guidelines (WCAG) για τον Παγκόσμιο Ιστό. Λίγο πολύ, οι αρχές που πρέπει να υιοθετηθούν, ούτως ή άλλως σε ένα προϊόν λογισμικού, ώστε αυτό να καθίσταται προσβάσιμο, είναι οι ίδιες.

Το καινούργιο όμως που υπάρχει όταν αναπτύσσεται μια εφαρμογή για το Mobile Device, είναι η κατάλληλη (και διαφορετική σε σχέση με το INTERNET) τεχνολογία και εργαλεία υλοποίησης κι η απαραίτητη προσαρμογή του περιεχομένου σε μικρές οθόνες. Η δυνατότητα δημιουργίας μιας τέτοιας εφαρμογής κι η εναπόθεση όλων των στοιχείων-αντικειμένων σε μικρή οθόνη, ώστε η πλοήγηση να καθίσταται λειτουργική, εύχρηστη, σαφής κι ευχάριστη είναι ένα στοίχημα.

Για την παρούσα εφαρμογή συγκεκριμένα, υιοθετήθηκαν οι εξής αρχές προσβασιμότητας:

- i. Εύχρηστο κι ευχάριστο περιβάλλον.
- ii. Σωστή αντίθεση χρωμάτων.
- iii. Σαφής κι εύκολη πλοήγηση, όμοια από φόρμα σε φόρμα.
- iv. Εναλλακτικός τρόπος πλοήγησης (πέρα από το βασικού μενού, υπάρχουν και σύνδεσμοι ή κουμπιά όπου είναι αναγκαίο).
- v. Κοινή και λογική διάρθρωση περιεχομένου από φόρμα σε φόρμα.
- vi. Τίτλοι σε κάθε φόρμα- σελίδα, σχετικοί με το περιεχόμενό της.
- vii. Επικεφαλίδες- τίτλοι για κάθε κείμενο και εικόνα.
- viii. Ορισμός συγκεκριμένου μεγέθους γραμματοσειρών για τους τίτλους, ανάλογα με τη σημαντικότητα των πληροφοριών του κειμένου που ορίζουν.
- ix. Σχετική με τη χρήση τους ονομασία των objects (= αντικειμένων).

- x. Απουσία σπασμένων links (= συνδέσμων).
- xi. Χρήση tags (= ετικετών) σε όλα τα αντικείμενα της εφαρμογής ώστε να γίνεται συνοπτική επεξήγηση του περιεχομένου και της χρήσης τους από τους screen readers.
- xii. Παροχή δυνατότητας αύξησης- μείωσης της γραμματοσειράς του κειμένου για να βοηθούνται στην ανάγνωση τα άτομα με προβλήματα όρασης.
- xiii. Δυνατότητα ηχητικής αναπαραγωγής του κειμένου για άτομα με προβλήματα όρασης.
- xiv. Κάθε ηχογράφιση κειμένου, διαθέτει ενσωματωμένους δύο επιπλέον ήχους ειδοποίησης για την αρχή και το τέλος της αναπαραγωγής.

Τα παραπάνω γνωρίσματα, αποδίδουν έναν ικανοποιητικό βαθμό προσβασιμότητας για μία τεχνολογία που επειδή είναι σχετικά νέα ακόμη, δεν έχει θέσει οι δημιουργοί της και σε πρώτη προτεραιότητα την παροχή προσβασιμότητας στους χρήστες (βέβαια τα δεδομένα αλλάζουν από μέρα σε μέρα), σε αντίθεση με το INTERNET και τις desktop εφαρμογές, όπου έχουν γίνει τεράστια βήματα τα τελευταία χρόνια στον τομέα αυτό.

### 3.8 Τεχνικές και μέθοδοι για την ικανοποίηση απαιτήσεων προσβασιμότητας

Όταν θέλουμε να ικανοποιούνται προϋποθέσεις προσβασιμότητας σε ένα προϊόν λογισμικού, υπάρχουν συγκεκριμένες αρχές που πρέπει να τηρηθούν. Πολλές από αυτές αναφέρθηκαν και στην προηγούμενη ενότητα. Το WCAG γενικά, έχει καθορίσει έναν καλό τρόπο κατηγοριοποίησης των κατευθυντήριων οδηγιών για την προσβασιμότητα σε A- AA- AAA, ανάλογα με τη σημαντικότητά τους (και τη δυσκολία ενσωμάτωσης στο site στις περισσότερες των περιπτώσεων). Από το A έως το AA, οι οδηγίες γίνονται λιγότερο σημαντικές.

Στην εφαρμογή που παρουσιάζεται για το Windows Mobile, όπως είναι λογικό, δόθηκε προτεραιότητα στις οδηγίες A ως περισσότερο σημαντικές και χρήσιμες συνολικά για τους χρήστες και ειδικότερα για τα ΑμεΑ. Παρακάτω γίνεται εξηγείται πως οι προϋποθέσεις που ικανοποιεί η εφαρμογή τηρήθηκαν. Για τις αρχές βεβαίως i, ii, x και xiv όπως αριθμούνται από την παραπάνω ενότητα, δε χρειάζεται να γίνει κάποια ειδική αναφορά:

**iii, iv:** Βάση της πλοήγησης, αποτελεί η χρήση φορμών. Η μετάβαση από φόρμα σε φόρμα γίνεται μέσω κουμπιών ή εικόνων και menu items που λειτουργούν ως κουμπιά, με αντίστοιχη ή εννοούμενη ονομασία για τη φόρμα μετάβασης. Ο κώδικας που χρησιμοποιείται για να γίνει αυτή η μετάβαση, ενδεικτικά είναι ο εξής:

```
private void EnterButton_Click(object sender, EventArgs e)
{
    ChoicesPage ChoicesPage = new ChoicesPage();
    ChoicesPage.Show();
    this.Hide();
}
```

Το πρόγραμμα δημιουργεί από μόνο του τη μέθοδο του event (= ενέργεια) του κουμπιού (εδώ το event είναι το κλικ του ποντικιού), ώστε να εισαχθεί εντός της η λειτουργία που αυτό θα πραγματοποιήσει. Στο συγκεκριμένο παράδειγμα, έχουμε διαδοχικά ανά σειρά τη δημιουργία νέου αντικειμένου για τη φόρμα προορισμού, την

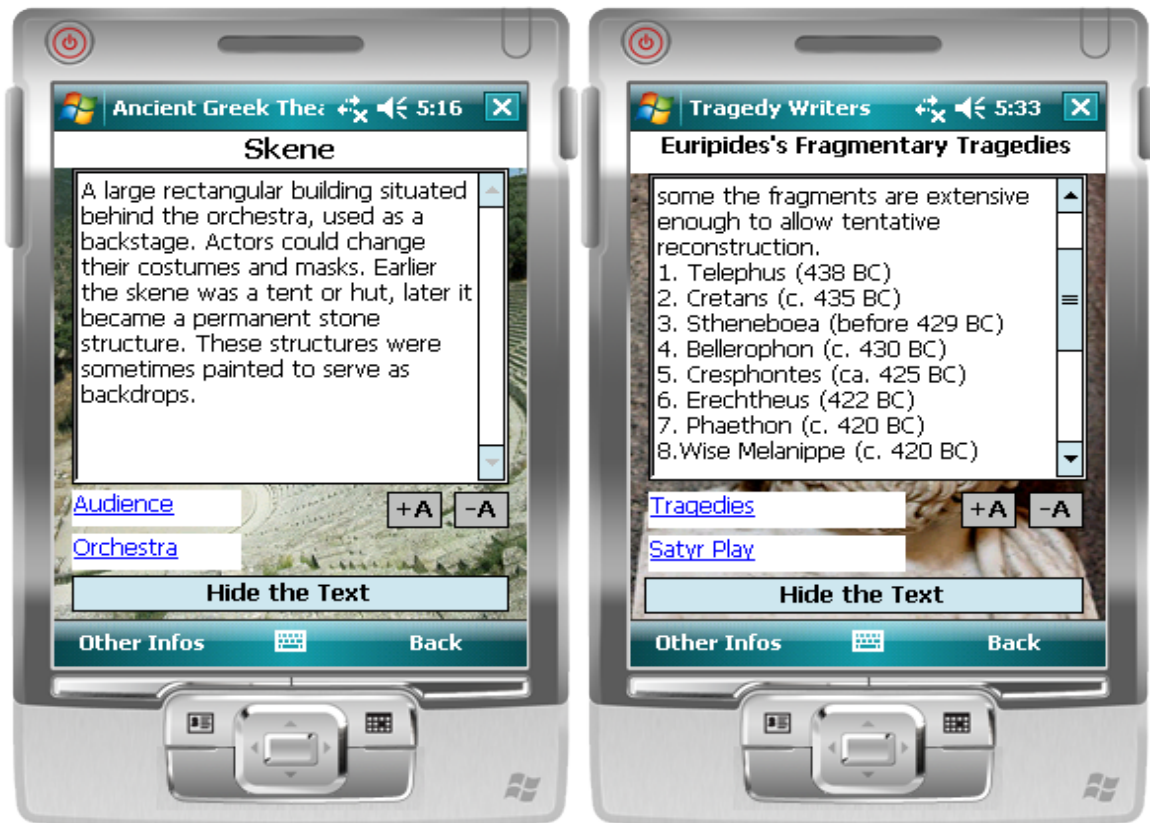
κλήση της μεθόδου Show(); για την εμφάνιση της νέας φόρμας και την απόκρυψη της φόρμας στην οποία βρίσκεται ο χρήστης τη δεδομένη στιγμή.

Όσον αφορά στη βασική πλοήγηση που πραγματοποιείται με τη βοήθεια main menu, ισχύει ότι ανάλογα με τα στοιχεία που είναι εμφανισμένα στη φόρμα, μπορεί να θέλουμε να καταστήσουμε ενεργά ή ανενεργά, να αλλάξουμε το όνομα και τις ιδιότητες κάποιου menu item, να προσθέσουμε ή να αφαιρέσουμε ένα από αυτά. Για παράδειγμα στη φόρμα Tragedy Writers, πριν γίνει η επιλογή κάποιου ποιητή στη λίστα επιλογής δε θέλουμε να είναι ενεργό κάποιο menu item. Μετά την επιλογή να είναι αυτή να είναι. Ανάλογα επίσης με τον κάθε ποιητή που θα επιλεγθεί, να αλλάζει η ονομασία (εννοούμε το text του menu item κι όχι το όνομα αυτού του αντικειμένου). Οι συγκεκριμένες αλλαγές βέβαια, αφορούν στην αλλαγή των ιδιοτήτων τους (properties), που αρχικά καθορίζονται από το Property Windows. Τα παραπάνω ενδεικτικά γίνονται με τον εξής τρόπο:

```
LifeText.Enabled = false; //Αφαίρεση της δυνατότητας επιλογής του
MenuItem με το όνομα LifeText
LifeText.Enabled = true; //Απόκτηση της δυνατότητας επιλογής του
MenuItem με το όνομα LifeText
WorksText.MenuItems.Remove(WorkCategory1); //Διαγραφή του menu item που
υπάρχει by default από το σχεδιασμό με το όνομα WorkCategory1
WorksText.MenuItems.Add(WorkCategory1); //Δημιουργία- προσθήκη MenuItem
με το όνομα WorkCategory1 με parent το MenuItem WorksText
WorkCategory1.Text = "Tragedies"; //Αλλαγή του της ιδιότητας (property)
Text του MenuItem WorkCategory1
```

Η εναλλακτική πλοήγηση γίνεται με τη βοήθεια κουμπιών και συνδέσμων. Ο τρόπος λειτουργίας τους και ενσωμάτωσης σε αυτά εντολών κώδικα για να καθοριστεί το τι θα κάνουν, είναι πάλι ακριβώς ο ίδιος. Σύνδεσμοι υπάρχουν στις περιπτώσεις όπου το περιεχόμενο της φόρμας εμφανίζεται μετά από το κλικ ενός menu item της δεύτερης σειράς αυτών και χρησιμοποιούνται για μπορεί να γίνεται άμεση μετάβαση στα άλλα της ίδιας σειράς. Το ρόλο επίσης εναλλακτικής πλοήγησης παίζουν, όπως ήδη έχουμε δει, και εικόνες που λειτουργούν ως κουμπιά (στη φόρμα Choices Page). Κι εδώ, οι αλλαγές των ιδιοτήτων ανάλογα με τις επιλογές του χρήστη κι όπου γενικά χρειάζεται,





Εικόνες 3-5/ 3-6: Screenshots φορμών με εμφάνιση του συνόλου των αντικειμένων τους

γίνονται με παρόμοιο τρόπο όπως με τα menu items. Πχ. για ένα συνδέσμο όταν γίνεται κλικ σε αυτόν μπορεί να έχουμε:

```

/*Χρήση εντολής if για τη λήψη απόφασης.
*Ο έλεγχος αφορά το Text του LinkLabel FirstLinkLabel
*Ανάλογα με το Text, θα εμφανίζονται τα παρακάτω αντικείμενα με
συγκεκριμένες ιδιότητες
* που αφορούν το συγκεκριμένο θέμα (του link)
*/
if (FirstLinkLabel.Text == "Audience")
{
    FirstLinkLabel.Text = "Skene"; //Αλλαγή Text στο LinkLabel
    FirstLinkLabel
    SecondLinkLabel.Visible = true; //Εμφάνιση του LinkLabel
    SecondLinkLabel
}

```

**Υ.:** Η κοινή και λογική διάρθρωση του περιεχομένου έγινε με σκοπό τον άμεσο κι εύκολο εγκληματισμό του χρήστη στην εφαρμογή, όπως και την εύκολη πλοήγησή του. Οι έντονες διαφοροποιήσεις περιεχομένου από φόρμα σε φόρμα, είναι μη πρακτικές και κουράζουν γενικά το χρήστη. Η προσαρμογή του στη λειτουργία της εφαρμογής γίνεται δυσκολότερη. Έτσι, έγινε προσπάθεια να κρατηθούν όλα σε μια λογική σειρά, το περιεχόμενο να παρουσιάζεται με τον ίδιο σχεδόν τρόπο σε όλες τις φόρμες και με τη σειρά που πρέπει να διαβάζεται (τουλάχιστον από το μάτι). Η κοινή διάρθρωση φαίνεται στις δύο παρακάτω εικόνες:



**vi.:** Σε κάθε φόρμα, υπάρχουν τίτλοι στο top bar σχετικοί με το περιεχόμενό της. Μπορεί να διακριθεί στην κορυφή των παραπάνω εικόνων.

**vii.:** Οι επικεφαλίδες κι οι τίτλοι για κάθε κείμενο και εικόνα, κρίνονται απαραίτητοι για να γνωρίζει ο χρήστης ανά πάσα στιγμή και με ματιά που βρίσκεται και τι είναι αυτό περιγράφουν τα στοιχεία και μεγαλύτερα σε μέγεθος στοιχεία της φόρμας (text box-κείμενο και εικόνα).



**Εικόνα 3-7:** Χρήση τίτλων για την διάρθρωση του περιεχομένου και την ονομασία των βασικών αντικειμένων της φόρμας

**viii.:** Ο καθορισμός του μεγέθους της γραμματοσειράς για τους τις επικεφαλίδες, είναι χρήσιμος για να δίνεται μία ιδέα της ιεραρχικής διάρθρωσης του περιεχομένου. Έτσι, το βασικό κείμενο σε κάθε φόρμα έχει μέγεθος γραμματοσειράς 13, οι υπόλοιποι τίτλοι που αντιστοιχούν στα πρώτης σειράς menu items μέγεθος 11 κι αυτοί που αντιστοιχούν στα δεύτερης σειράς menu items έχουν μέγεθος 10. Το default μέγεθος γραμματοσειράς στα labels (που ως tools παίζουν το ρόλο επικεφαλίδας- τίτλων), όπως βέβαια και του στυλ και του τύπου γραμματοσειράς που δε μας ενδιαφέρει ιδιαίτερα παρά μόνο για να είναι ευνόητοι κι ευκολοδιάβαστοι οι τίτλοι, καθορίζονται από το Property Window του κάθε label. Για την αλλαγή των τιμών που έχουν καθοριστεί ως default, χρησιμοποιείται ο παρακάτω κώδικας:

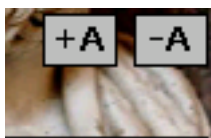
```
TextTitleLabel.Font = new Font("Tahoma", 11, FontStyle.Bold);  
//Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label  
TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος  
γραμματοσειράς, στυλ γραμματοσειράς
```

**ix.:** Η σχετική με τη σκοπιμότητα και την παρεχόμενη πληροφορία των αντικειμένων, είναι σημαντική για δύο κύρια λόγους. Αρχικά, από την πλευρά του προγραμματιστή να γνωρίζει τι κάνει το κάθε αντικείμενο και να το εντοπίζει εύκολα κατά τη συγγραφή κώδικα. Η τυπική, πρόχειρη κι απρόσεκτη ονομασία αντικειμένων με όμοια λειτουργία, όπως η αρίθμηση τους, γίνεται ένας κακός μπελάς όπου δε βγαίνει άκρη στον προγραμματισμό, ειδικά όταν ο κώδικας μεγαλώνει σε μέγεθος. Δεύτερον, όταν οι screens readers «διαβάζουν» ένα αντικείμενο, ο χρήστης πρέπει να κατανοεί ακριβώς αυτό τι κάνει, ποιος είναι ο ρόλος του μέσα στην εφαρμογή. Σε διαφορετική περίπτωση, η αρχή της καλής σχεδίασης, τουλάχιστον σε αυτόν τον τομέα, έχει αποτύχει.

**xi.:** Η χρήση ετικετών είναι βαρύνουσας σημασίας για την αρχή της καλής σχεδίασης στον τομέα της ανάπτυξης εφαρμογών. Οι ετικέτες είναι ένα μεταδεδομένο που περιέχει μία συνοπτική πληροφορία για ένα αντικείμενο, ώστε αυτό να εντοπίζεται από εσωτερική ή εξωτερική αναζήτηση. Είναι επίσης σημαντικό γιατί αποδίδει με μία λέξη κλειδί ή μια μικρή φράση τη σκοπιμότητά του.

Στην εφαρμογή, τα tags ορίζονται για τα διάφορα στοιχεία μέσα από το Property Window. Ωστόσο, επειδή το ίδιο αντικείμενο αλλάζει συνεχώς πληροφορία (η χρήση περιορισμένων αντικειμένων, βοηθούν γενικά στη σχεδίαση), πρέπει ανάλογα με αυτή, να αλλάζει και το tag της. Αυτό πραγματοποιείται με τον εξής τρόπο:

```
TextTitleLabel.Tag = "Title- Euripides's \"Satyr Play\""; //Αλλαγή του  
Tag του Label TextTitleLabel
```



**Εικόνα 3-8:** Κουμπιά για την αύξηση- μείωση του μεγέθους γραμματοσειράς

**xii.:** Μια ιδιαίτερα χρήσιμη για την προσβασιμότητα μέθοδος διευκόλυνσης των ατόμων με προβλήματα όρασης, είναι αυτή της αύξησης και της μείωσης του μεγέθους γραμματοσειράς των γραμμάτων ενός κειμένου. Με αυτόν τον τρόπο παρέχεται η δυνατότητα στον ίδιο το χρήστη να καθορίσει το μέγεθος γραμματοσειράς που τον βολεύει για να μπορεί να διαβάσει το κείμενο. Στην παρούσα εφαρμογή, η παραπάνω ιδιότητα πραγματοποιείται με τη βοήθεια δύο κουμπιών, όπως φαίνεται στην παρακάτω εικόνα.

Ο κώδικας που απαιτείται για εδώ, είναι ο εξής:

```
private void increaseTextButton_Click(object sender, EventArgs e)
{
    float k = TragedyWritersText.Font.Size; /*Δήλωση μεταβλητή
*για το Size της Font του TextBox TragedyWritersText, τύπου float.
*Η αρχική τιμή είναι αυτή που έχει ορισθεί by default δλδ = 9em.
*/
    if (k <= 18) //Εντολή απόφασης για τον έλεγχο της τιμής του k
    {
```

```

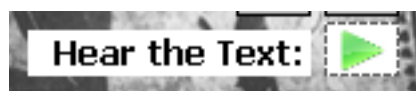
        /*Κατασκευή αντικειμένου για το Font του TextBox
*BasicInformationText.
        * Στη 2η παράμετρο (fontSize) γίνεται αύξηση κατά 1em στο
*υπάρχον Size της Font του TextBox TragedyWritersText
*Οι υπόλοιπες ιδιότητες, παραμένουν ως έχουν by default από το σχεδιασμό
*/
        TragedyWritersText.Font = new Font("Tahoma", k + 1,
FontStyle.Regular);
    }

private void decreaseTextButton_Click(object sender, EventArgs e)
{
    float k = TragedyWritersText.Font.Size; /*Δήλωση μεταβλητή
*για το Size της Font του TextBox TragedyWritersText, τύπου float.
*Η αρχική τιμή είναι αυτή που έχει ορισθεί by default δλδ = 9em.
*/
    if (k >= 8) //Εντολή απόφασης για τον έλεγχο της τιμής του k
    {
        /* Κατασκευή αντικειμένου για το Font του TextBox
*BasicInformationText.
        * Στη 2η παράμετρο (fontSize) γίνεται μείωση κατά 1em στο
*υπάρχον Size της Font του TextBox TragedyWritersText
*Οι υπόλοιπες ιδιότητες, παραμένουν ως έχουν by default από το σχεδιασμό
*/
        TragedyWritersText.Font = new Font("Tahoma", k - 1,
FontStyle.Regular);
    }
}

```

**xiii.:** Κάτι άλλο που θέτει ως οδηγία το WCAG, είναι η δυνατότητα ηχητικής αναπαραγωγής του κειμένου ως εναλλακτική πρόσβαση. Τα άτομα που έχουν σοβαρό πρόβλημα όρασης ή ολική τύφλωση, δεν μπορεί να αρκестούν στην αύξηση του μεγέθους γραμματοσειράς. Για να μάθουν το περιεχόμενο ενός πλαισίου κειμένου, πρέπει να αυτό γίνει με διαφορετικό από την ανάγνωση τρόπο. Ένας τέτοιος τρόπος είναι ο ήχος.

Στην εφαρμογή, αυτή η δυνατότητα ηχητικής αναπαραγωγής του κειμένου, δίνεται μέσω ενός κουμπιού, όπως αντίστοιχα κι η διακοπή αναπαραγωγής. Τα κουμπιά αυτά έχουν διαμορφωθεί με κατάλληλη τρόπο ώστε να παραπέμπουν άμεσα στη λειτουργία τους ως Play και Stop buttons. Πέρα από την ονομασία τους που δηλώνει ακριβώς αυτό, υπάρχει κι ένα label στα αριστερά τους, που με το κείμενό του προσδιορίζει το σκοπό τους. Όλα αυτά φαίνονται στην επόμενη εικόνα.



**Εικόνα 3-9:** Κουμπί αναπαραγωγής ήχου και κείμενο για τον προσδιορισμό της λειτουργίας του. Όταν το PlayButton πατηθεί, αυτόματα αλλάζει σε StopButton με την κατάλληλη εμφάνιση

Ο κώδικας που απαιτείται για εδώ, είναι ο εξής:

```

using System.Media; /*Namespace που εισάγει στην εμβέλεια του κώδικα της
*φόρμας τις μεθόδους για τα πολυμέσα
*/

```

```

using System.IO; //Namespace που εισάγει στην εμβέλεια του κώδικα της
*μεθόδους για την είσοδο- έξοδο πληροφοριών
*/

MediaPlayer Media = new MediaPlayer(); /*Κατασκευή αντικειμένου
*προγράμματος αναπαραγωγής ήχου με το όνομα Media
*Τοποθετείται μέσα στη γενική κλάση της φόρμας*/

private void PlayButton_Click(object sender, EventArgs e) /*Κατασκευή
*μεθόδου για το κλικ του κουμπιού PlayButton
{
    PlayButton.Visible = false; //Απόκρυψη του button PlayButton
    StopButton.Visible = true; //Εμφάνιση του button StopButton
    Stream STR = new MemoryStream(Properties.Resources.Tragedy);
/*Κατασκευαστής ρόης ήχου με το όνομα STR και άντληση αυτού από το source
*/
    Media = new MediaPlayer(STR); /*Κατασκευαστής προγράμματος
*αναπαραγωγής ήχου και ανάθεση αυτού στην τιμή Media
*/
    Media.Play(); //Αναπαραγωγή του ήχου
}

private void StopButton_Click(object sender, EventArgs e)
/*Κατασκευή μεθόδου για το κλικ του κουμπιού StopButton
*/
{
    PlayButton.Visible = true; //Εμφάνιση του button PlayButton
    StopButton.Visible = false; //Απόκρυψη του button StopButton
    Stream STR = new MemoryStream(Properties.Resources.Tragedy);
/*Κατασκευαστής ρόης ήχου με το όνομα STR και άντληση αυτού από το source
*/
    Media = new MediaPlayer(STR); /*Κατασκευαστής προγράμματος
*αναπαραγωγής ήχου και ανάθεση αυτού στην τιμή Media
*/
    Media.Stop(); //Διακοπή αναπαραγωγής του ήχου
}

```

### 3.9 Αξιοποίηση των δυνατοτήτων της εφαρμογής

Η εφαρμογή παρέχει συγκεκριμένες δυνατότητες τις οποίες ο χρήστης καλείται να αξιοποιήσει. Αν και οι περισσότερες έχουν ήδη αναφερθεί καθώς ήταν σχετικές με την πλοήγηση και την προσβασιμότητα, εδώ θα δούμε κάποιες άλλες βασικές για την εμφάνιση του πληροφοριακού περιεχομένου. Το ενδιαφέρον θα εστιαστεί στις φόρμες 3, 4 και 5 που το διαθέτουν. Επίσης θα δοθούν και μικρά παραδείγματα κώδικα για να γίνει κατανοητή η ανάπτυξη των δυνατοτήτων.

Μπαίνοντας στις δύο πρώτες φόρμες με το περιεχόμενο (Basic Information και Ancient Greek Theater), ο χρήστης θα δει τη σχετική κεντρική εικόνα, τον τίτλο του θέματός της και γενικά της φόρμας και ένα κουμπί με κείμενο πάνω του που να αναγράφει «Read the Text». Κάνοντας κάποιος κλικ σε αυτό το κουμπί θα δει να ανοίγεται το text box που περιέχει το κείμενο του θέματος πάντα στο οποίο αναφέρεται ο τίτλος στην κορυφή της φόρμας. Επίσης, θα εμφανιστούν τα κουμπιά και για την αύξηση- μείωση του μεγέθους γραμματοσειράς και της αναπαραγωγής του ήχου (το Play button μαζί με το σχετικό βέβαια label). Ταυτόχρονα, όμως το κουμπί Read The Text θα αλλάξει και θα μετατραπεί σε ένα άλλο που θα αναγράφει «Hide the Text». Αυτό φυσικά όταν πατηθεί θα φέρει την

εφαρμογή στην αρχική της εμφάνιση, πέραν του τίτλου πάνω στη φόρμα που θα παραμείνει ως έχει (για το λόγο που στην περίπτωση πατήσει εκ νέου το Read the Text button ο χρήστης).



**Εικόνες 3-10 /3-11:** Κουμπιά Read the Text και Hide the Text

Βέβαια, όπως έχει γραφτεί και προηγούμενα, πρόσβαση στα υπόλοιπα κείμενα και στα αντικείμενα που το συνοδεύουν, αλλά και στο ίδιο αρχικό- βασικό κείμενο, γίνεται και μέσω των δύο τρόπων πλοήγησης. Σε αυτή την περίπτωση θα παρακαμφθεί η λειτουργία του Read the Button, διότι αυτή θα έχει πραγματοποιηθεί από τα menu items ή τα links. Έπειτα όμως, το Hide the Text button θα έχει την ίδια ακριβώς λειτουργία, δεν αλλάζει κάτι σε σχέση με προηγούμενα. Παρακάτω, παρατίθενται από ένα παράδειγμα συγγραφής κώδικα για καθένα από αυτά τα δύο κουμπιά.

#### **Read the Text Button:**

```
private void ReadTheTextButton_Click(object sender, EventArgs e)
//Μέθοδος για το Event Click του button ReadTheTextButton
{
    AncientGreekTheaterText.Visible = true; /*Εμφάνιση του
*TextBox AncientGreekTheaterText
*/
    ReadTheTextButton.Visible = false; /*Απόκρυψη του Button
*ReadTheTextButton
*/
    HideTheTextButton.Visible = true; /*Εμφάνιση του Button
*HideTheTextButton
*/
    increaseTextButton.Visible = true; /*Εμφάνιση του button
*increaseTextButton
*/
    decreaseTextButton.Visible = true; /*Εμφάνιση του button
*decreaseTextButton
*/
    /*
    * Χρήση εντολής if για τη λήψη απόφασης
    * για τον έλεγχο του Text του Label TextTitleLabel (έλεγχος
*για το αν ισχύει μία από τις τρεις αναγραφόμενες περιπτώσεις)
    * και για το visibility του TextBox AncientGreekTheaterText
* (έλεγχος για το αν αυτό θα είναι εμφανές)
    * Για να μπει το πρόγραμμα στον έλεγχο πρέπει να ισχύουν και
*τα δύο παραπάνω ώστε να εκτελεστούν οι ενέργειες που βρίσκονται μέσα σε
*αυτόν και αφορούν το visibility των LinkLabels που υπάρχουν στη φόρμα
    */
    if (((TextTitleLabel.Text == "Audience") ||
(TextTitleLabel.Text == "Skene") || (TextTitleLabel.Text == "Orchestra"))
&& (AncientGreekTheaterText.Visible == true))
    {
        FirstLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*FirstLinkLabel
*/
    }
}
```

```

                SecondLinkLabel.Visible = true;    /*Απόκρυψη του LinkLabel
*SecondLinkLabel
*/
            }
        }
    }
}

```

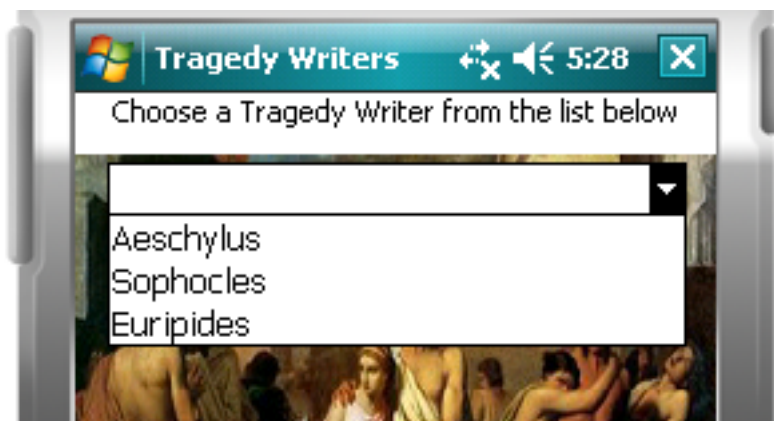
### **Hide the Text Button:**

```

private void HideTheTextButton_Click(object sender, EventArgs e)
{
    AncientGreekTheaterText.Visible = false;    /*Απόκρυψη του
*TextBox BasicInformationText
*/
    ReadTheTextButton.Visible = true;    /*Εμφάνιση του Button
*ReadTheTextButton
*/
    HideTheTextButton.Visible = false;    /*Απόκρυψη του Button
*HideTheTextButton
*/
    FirstLinkLabel.Visible = false;    /*Απόκρυψη του LinkLabel
*FirstLinkLabel
*/
    SecondLinkLabel.Visible = false;    /*Απόκρυψη του LinkLabel
*SecondLinkLabel
*/
    increaseTextButton.Visible = false;    /*Απόκρυψη του button
*increaseTextButton
*/
    decreaseTextButton.Visible = false;    /*Απόκρυψη του button
*decreaseTextButton
*/
}

```

Τα πράγματα αλλάζουν κάπως στην τελευταία 5<sup>η</sup> φόρμα με τον τίτλο Tragedy Writers. Εδώ, χρησιμοποιείται αρχικά ένα ComboBox (λίστα επιλογής) με τρεις προκαθορισμένες και σταθερές επιλογές, τις Aeschylus, Sophocles κι Euripides. Ανάλογα με την επιλογή που θα κάνει ο χρήστης, θα εμφανιστεί κι η αντίστοιχη εικόνα για τον τραγικό ποιητή και το κουμπί Read the Text που κι εδώ επιτελεί τον ίδιο με τις άλλες φόρμες ρόλο. Ο τίτλος επίσης αρχικά στην φόρμα, προτρέπει το χρήστη να κάνει μία επιλογή στο ComboBox. Εφόσον γίνει μία επιλογή κι αυτός επιτελεί το γνωστό του ρόλο. Ο κώδικας του ComboBox παρατίθεται στο παράρτημα στη σελ. 83.



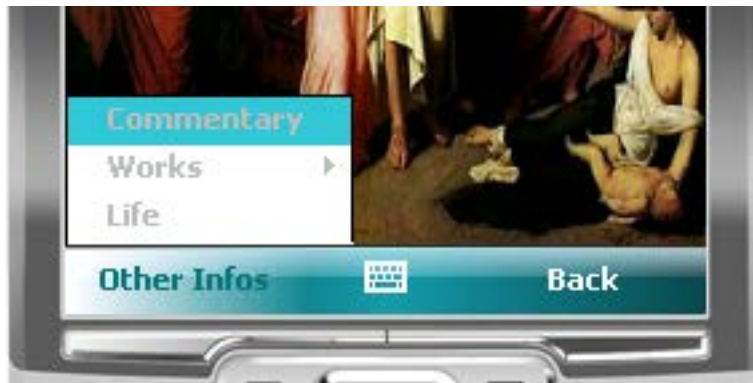


**Εικόνα 3-12:** Το ComproBox (λίστα επιλογής) σε αναδίπλωση

Στο κουμπί Hide the Text εδώ, επανεμφανίζεται η φόρμα, όπως είναι αρχικά κατά την είσοδο του χρήστη, δηλαδή με το ComproBox και την αρχική εικόνα.

Επαναλαμβάνεται τέλος, πως στη συγκεκριμένη φόρμα, όταν βρισκόμαστε στην αρχική κατάστασή της, τα menu items είναι μη ενεργά (πέρα από το menu item back), καθώς για να αποδώσουν σε περίπτωση πατήματός τους τα κατάλληλα στοιχεία, πρέπει πρώτα να επιλεγεί ένας ποιητής από το ComproBox. Επίσης, η ύπαρξη και το text των menu items δεύτερης σειράς όπως και το πιο από αυτά της πρώτης δείχνει σε εκείνα, καθορίζονται πάλι από την επιλογή που θα γίνει στο Comprobox. Όταν γίνει αυτή, όλα τα menu items ενεργοποιούνται κανονικά και μπορούν να χρησιμοποιηθούν με το γνωστό τρόπο, κι όταν πάλι πατηθεί το κουμπί Hide the Text, απενεργοποιούνται εκ νέου. Τα ανενεργά menu items, καθορίζονται γενικά με την εντολή:

```
MenuItem1.Enabled = false;
```



**Εικόνα 3-13:** Ανενεργά menu items

Επειδή πάντως γενικά τα αντικείμενα είναι πολλά στις φόρμες, το αν αυτά θα εμφανίζονται ή όχι καθορίζεται από το visibility τους (= ορατότητα ή πιο σωστά εμφάνισή τους). Το αν τώρα αυτά θα εμφανίζονται ή όχι και πότε θα συμβαίνει αυτό, καθορίζεται χονδρικά από τις παραμέτρους που προαναφέρθηκαν αλλά και από άλλους (για παράδειγμα, αν είναι ενεργοποιημένη ή όχι η φόρμα, ποιο είναι το κείμενο που αναγράφεται στον τίτλο κλπ).

# Κεφάλαιο 4: Αξιολόγηση Εφαρμογής

## 4.1 Αποτελέσματα- Αξιολόγηση

Το τελικό project που παρουσιάζεται, πληρεί το σύνολο σχεδόν από τις απαιτήσεις και τους στόχους που τέθηκαν κατά το σχεδιασμό, και στην πορεία και βάση του μοντέλου που χρησιμοποιήθηκε, κατά την ανάπτυξη του λογισμικού. Έτσι λοιπόν εκπληρώνει και εξυπηρετεί με επιτυχία συγκεκριμένους στόχους κι η ίδια αυτή καθαυτή η εφαρμογή, αλλά και ως λειτουργικό Windows Mobile, με την έννοια ότι αποδεικνύεται πράγματι σε έναν ικανοποιητικό βαθμό η δυνατότητα ενσωμάτωσης οδηγιών προσβασιμότητας στο τελευταίο. Πιο συγκεκριμένα επιτυγχάνονται:

- Η κατά το δυνατόν ισότιμη παρουσίαση του περιεχομένου της εφαρμογής σε κοινούς χρήστες και σε ΑμεΑ.
- Η πειθάρχηση με διεθνή standards προσβασιμότητας όπως ορίζονται από τον οργανισμό Web Accessibility Initiative, προσαρμοσμένες κατάλληλα σε μικρή οθόνη. Η εφαρμογή σημαντικού μέρους βασικών οδηγιών όπως ορίζονται από τον Web Content Accessibility Guidelines.
- Η βέλτιστη αξιοποίηση των γνώσεων που αποκτήθηκαν με τη χρήση των εργαλείων υλοποίησης μέσα σε ένα χρόνο, για το θετικότερο δυνατό αποτέλεσμα.
- Η σωστή χρησιμοποίηση κώδικα και των εργαλείων υλοποίησης που αφήνει περιθώρια για εύκολη βελτίωση- αναβάθμιση της εφαρμογής στο μέλλον και επαναπροσδιορισμό κάποιων ανώτερα εξειδικευμένων στόχων.
- Η ευχάριστη τελικά εμπειρία πλοήγησης από μέρους του χρήστη!

Αναφέρθηκε ήδη πώς βάση και πρότυπο για την ενσωμάτωση προϋποθέσεων προσβασιμότητας στην εφαρμογή, ήταν οι οδηγίες του WCAG. Βέβαια, πολλές από αυτές που είναι απλές στην εφαρμογή τους δε χρησιμοποιήθηκαν γιατί δεν υπήρχαν τα στοιχεία στο λογισμικό στα οποία αναφέρονται (πχ. κανόνες ενσωμάτωσης προσβασιμότητας για τη συμπλήρωση φορμών, συγχρονισμένα πολυμέσα κλπ.). Το θετικό βέβαια, είναι ότι οι αρχικοί στόχοι που μπήκαν ικανοποιήθηκαν, άλλοι σε μεγαλύτερο κι άλλοι σε μικρότερο βαθμό σε σύγκριση με τα αρχικά σχέδια. Σημασία έχει πάντως πως το τελικό αποτέλεσμα ικανοποιεί ως ένα βαθμό το δημιουργό.

Επειδή όμως από το καλό, υπάρχει πάντα το καλύτερο κι επειδή, ειδικά η τεχνολογία λογισμικού, είναι ένα κλάδος όπου η συνεχής βελτίωση των προϊόντων που παράγονται και για την ικανοποίηση νέων διευρυμένων αναγκών αλλά και για τη διόρθωση λαθών, είναι απαραίτητη, μεριμνήθηκε ώστε να παρέχεται εύκολα αυτή η δυνατότητα. Ο πιο ασφαλής βεβαίως τρόπος γι' αυτό ήταν η συγγραφή νοικοκυρεμένου κώδικα (αν και το ίδιο το Visual Studio ενισχύει και βοηθάει σε αυτή την προσπάθεια) και η ύπαρξη σχολίων.

## 4.2 Συμπεράσματα

Για τον Παγκόσμιο Ιστό, έχει μεριμνηθεί πολύ σωστά κι έχει διατυπωθεί αρκετά εκτενώς και με σαφήνεια, πώς θα πρέπει να σχεδιάζονται, να δομούνται και να αναπτύσσονται τα sites για να ικανοποιούν προϋποθέσεις προσβασιμότητας. Μάλιστα, η συνέπεια αυτή από μέρους του μηχανικού λογισμικού, παροτρύνεται να «διαφημίζεται» κι από την αντίστοιχη πιστοποίηση που παρέχεται από τους σχετικούς με τη συγκεκριμένη έρευνα οργανισμούς και τις ιστοσελίδες τους. Η κατάληψη μεγαλύτερου μεριδίου αγοράς και η «καθαρή συνείδηση», από πλευράς δημιουργού, της προώθησης δημοκρατίας απέναντι στα ΑμεΑ, όπως και στους υπόλοιπους χρήστες, είναι το αντίκτυπο.

Τα συνεχή αυτά βήματα που γίνονται (ή και άλματα!) είναι πιο μεγάλα ακόμη κι από το αντίστοιχο επίπεδο των κλασικών λογισμικών για desktop. Το σίγουρο είναι, πως στις μέρες μας υπάρχει αυτή η ευαισθησία και γίνονται χρόνο με το χρόνο συνείδηση των μηχανικών λογισμικού οι αρχές της καλής σχεδίασης.

Υπάρχουν όμως και νέα δεδομένα. Πέρα από το γεγονός ότι η ίδια η έρευνα για την προσβασιμότητα πρέπει να ακολουθεί ή ακόμη και να ξεπερνά την εξέλιξη της τεχνολογίας του Τομέα των Πληροφοριών και των Επικοινωνιών, οι υπολογιστές πλέον δεν είναι μόνο οι Η/Υ ή τα laptops. Η ζήτηση για κινητά τηλέφωνα και Mobile Devices συνεχώς αυξάνεται. Το μεγαλύτερο μέρος των λειτουργιών μέσω INTERNET, αλλά και η χρήση κοινών λογισμικών, θα γίνεται σε λίγα χρόνια μέσω αυτών των συσκευών, που τείνουν να γίνουν ισχυροί υπολογιστές χειρός (αν δεν είναι ήδη).

Οι αρχές της καλής σχεδίασης πρέπει να ακολουθήσουν αυτή την τάση. Κι είναι γεγονός ότι όλο και μεγαλύτερο κομμάτι του επιστημονικού δυναμικού και εταιρειών λογισμικού στρέφουν το κύριο ενδιαφέρον τους προς αυτόν τον τομέα. Κι η πρόκληση εδώ είναι να γίνει ένα ακόμη παραπάνω βήμα στην προσβασιμότητα. Δηλαδή, οι νέες εφαρμογές που αναπτύσσονται να μην είναι απλά προσιτές από τα ΑμεΑ, αλλά πραγματικά χρηστικές.

Αν και το παρόν κείμενο παρουσιάζει μεθόδους και τεχνικές προσβασιμότητας που δε χαρακτηρίζονται από ιδιαίτερη πολυπλοκότητα και ξεχωριστό ή εντυπωσιακό σχεδιασμό, φιλοδοξεί ωστόσο να αποδείξει ότι αυτή μπορεί να επιτευχθεί και με τη σύγχρονη τάση της σμίκρυνσης του μεγέθους των Η/Υ ή της μετάβασης των λειτουργιών τους στις Mobile Devices. Τα εργαλεία υλοποίησης που αναλύθηκαν είναι κατάλληλα και χρήσιμα για το εγχείρημα αυτό, χωρίς σε καμία βεβαίως περίπτωση να είναι τα μοναδικά. Η διαφορά όμως της δυνατότητας ανάπτυξης προσβάσιμου υλικού για τους Η/Υ σε σχέση με την αντίστοιχη δυνατότητα για τις Mobile Devices, είναι ακόμη χαώδης, πέραν των άλλων και λόγω φυσικά της απόκλισης στην υπολογιστική ισχύ εις βάρος των δεύτερων. Το δεδομένο αυτό ίσως να μην αντιστραφεί και ποτέ.

Εκείνο όμως που θέλει να περάσει στον αναγνώστη και στον υποψήφιο ή τωρινό μηχανικό λογισμικού για κινητές συσκευές το κείμενο, είναι πως τελικά είναι πιο σημαντικές η καθολική πρόσβαση κι η χρησιμότητα που μια εφαρμογή για μία Mobile Device προωθεί, παρά η αισθητική πρόκληση.

# Παράρτημα

---

## Form1 – Welcome Page:

### Κώδικας ενεργειών:

```
using System;
using System.Linq;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Diploma_Work
{
    public partial class WelcomePage : Form
    {
        public WelcomePage()
        {
            InitializeComponent();

            private void EnterButton_Click(object sender, EventArgs e)
            //Μέθοδος για το event Click της Image EnterButton
            {
                ChoicesPage ChoicesPage = new ChoicesPage(); /*Κατασκευή
                *αντικειμένου ChoicesPage με εμβέλεια τοπική (μεθόδου)
                */
                ChoicesPage.Show(); //Εμφάνιση της φόρμας Choices Page
                this.Hide(); //Απόκρυψη της παρούσας φόρμας
            }
        }
    }
}
```

### Κώδικας σχεδιασμού:

```
namespace Diploma_Work
{
    partial class WelcomePage
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;
        private System.Windows.Forms.MainMenu mainMenu1;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
    }
}
```

```

    /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(WelcomePage));
    this.mainMenu1 = new System.Windows.Forms.MainMenu();
    this.EnterButton = new System.Windows.Forms.PictureBox();
    this.TitleImage = new System.Windows.Forms.PictureBox();
    this.TragicComicMasksImage = new
System.Windows.Forms.PictureBox();
    this.SuspendLayout();
    //
    // EnterButton
    //
    this.EnterButton.Image =
((System.Drawing.Image)(resources.GetObject("EnterButton.Image")));
    this.EnterButton.Location = new System.Drawing.Point(49, 234);
    this.EnterButton.Name = "EnterButton";
    this.EnterButton.Size = new System.Drawing.Size(145, 25);
    this.EnterButton.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
    this.EnterButton.Tag = "Enter button";
    this.EnterButton.Click += new
System.EventHandler(this.EnterButton_Click);
    //
    // TitleImage
    //
    this.TitleImage.Image =
((System.Drawing.Image)(resources.GetObject("TitleImage.Image")));
    this.TitleImage.Location = new System.Drawing.Point(-1, 0);
    this.TitleImage.Name = "TitleImage";
    this.TitleImage.Size = new System.Drawing.Size(241, 40);
    this.TitleImage.Tag = "Ancient Greek Tragedy Title Image";
    //
    // TragicComicMasksImage
    //
    this.TragicComicMasksImage.Image =
((System.Drawing.Image)(resources.GetObject("TragicComicMasksImage.Image")
));
    this.TragicComicMasksImage.Location = new
System.Drawing.Point(0, 38);
    this.TragicComicMasksImage.Name = "TragicComicMasksImage";
    this.TragicComicMasksImage.Size = new System.Drawing.Size(240,
230);
    this.TragicComicMasksImage.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;

```

```

        this.TragicComicMasksImage.Tag = "Tragic Comic Masks";
        //
        // WelcomePage
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(96F, 96F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Dpi;
        this.AutoScroll = true;
        this.ClientSize = new System.Drawing.Size(240, 268);
        this.Controls.Add(this.EnterButton);
        this.Controls.Add(this.TitleImage);
        this.Controls.Add(this.TragicComicMasksImage);
        this.Menu = this.mainMenu1;
        this.Name = "WelcomePage";
        this.Tag = "Welcome Page";
        this.Text = "Welcome Page";
        this.ResumeLayout(false);
    }

    #endregion

    private System.Windows.Forms.PictureBox TragicComicMasksImage;
    private System.Windows.Forms.PictureBox TitleImage;
    private System.Windows.Forms.PictureBox EnterButton;
}
}

```

## Form2 – Choices Page:

### Κώδικας ενεργειών:

```

using System;
using System.Linq;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Diploma_Work
{
    public partial class ChoicesPage : Form
    {
        public ChoicesPage()
        {
            InitializeComponent();
        }

        private void BasicInformationButton_Click(object sender, EventArgs
e) //Μέθοδος για το event Click της Image BasicInformationButton
        {
            BasicInformation BasicInformation = new BasicInformation();
            //Κατασκευή αντικειμένου BasicInformation με εμβέλεια τοπική (μεθόδου)
            BasicInformation.Show();
            //Εμφάνιση της φόρμας BasicInformation
        }
    }
}

```



```

        this.Hide();
//Απόκρυψη της παρούσας φόρμας
    }

    private void AncientGreekTheaterButton_Click(object sender,
EventArgs e) //Μέθοδος για το event Click του AncientGreekTheaterButton
    {
        AncientGreekTheater AncientGreekTheater = new
AncientGreekTheater(); /*Κατασκευή αντικειμένου AncientGreekTheater με
*εμβέλεια τοπική (μεθόδου)
*/
        AncientGreekTheater.Show(); /*Εμφάνιση της φόρμας
*AncientGreekTheater
*/
        this.Hide(); //Απόκρυψη της παρούσας φόρμας
    }

    private void TragedyWritersButton_Click(object sender, EventArgs
e) //Μέθοδος για το event Click του TragedyWritersButton
    {
        TragedyWriters TragedyWriters = new TragedyWriters();
//Κατασκευή αντικειμένου TragedyWriters με εμβέλεια τοπική (μεθόδου)
        TragedyWriters.Show(); //Εμφάνιση της φόρμας TragedyWriters
        this.Hide(); //Απόκρυψη της παρούσας φόρμας
    }

    private void BasicInformationMenuButton_Click(object sender,
EventArgs e) /*Μέθοδος για το event Click του MenuItem
*BasicInformationMenuButton (εναλλακτική πλοήγηση)
*/
    {
        BasicInformation BasicInformation = new BasicInformation();
//Κατασκευή αντικειμένου BasicInformation με εμβέλεια τοπική (μεθόδου)
        BasicInformation.Show();
//Εμφάνιση της φόρμας BasicInformation
        this.Hide();
//Απόκρυψη της παρούσας φόρμας
    }

    private void AncientGreekTheaterMenuButton_Click(object sender,
EventArgs e) /*Μέθοδος για το event Click του MenuItem
*AncientGreekTheaterMenuButton (εναλλακτική πλοήγηση)
*/
    {
        BasicInformation BasicInformation = new BasicInformation();
//Κατασκευή αντικειμένου AncientGreekTheater με εμβέλεια τοπική (μεθόδου)
        BasicInformation.Show(); /*Εμφάνιση της φόρμας
*AncientGreekTheater
*/
        this.Hide(); //Απόκρυψη της παρούσας φόρμας
    }

    private void BackButton_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem BackButton
    {
        WelcomePage WelcomePage = new WelcomePage(); /*Κατασκευή
*αντικειμένου WelcomePage με εμβέλεια τοπική (μεθόδου)
*/
        WelcomePage.Show(); //Εμφάνιση της φόρμας BasicInformation
        this.Hide(); //Απόκρυψη της παρούσας φόρμας
    }

```

```

        private void TragedyWritersMenuButton_Click(object sender,
EventArgs e) //Μέθοδος για το event Click του MenuItem TragedyWriters
        {
            TragedyWriters TragedyWriters = new TragedyWriters();
//Κατασκευή αντικειμένου TragedyWriters με εμβέλεια τοπική (μεθόδου)
            TragedyWriters.Show(); //Εμφάνιση της φόρμας TragedyWriters
            this.Hide(); //Απόκρυψη της παρούσας φόρμας
        }
    }
}

```

## Κώδικας σχεδιασμού:

```

namespace Diploma_Work
{
    partial class ChoicesPage
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;
        private System.Windows.Forms.MainMenu mainMenu1;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(ChoicesPage));
            this.mainMenu1 = new System.Windows.Forms.MainMenu();
            this.menuItem1 = new System.Windows.Forms.MenuItem();
            this.TragedyWritersMenuButton = new
System.Windows.Forms.MenuItem();
            this.AncientGreekTheaterMenuButton = new
System.Windows.Forms.MenuItem();
            this.BasicInformationMenuButton = new
System.Windows.Forms.MenuItem();
            this.BackButton = new System.Windows.Forms.MenuItem();
            this.TragedyWritersButton = new
System.Windows.Forms.PictureBox();

```

```

        this.AncientGreekTheaterButton = new
System.Windows.Forms.PictureBox();
        this.BasicInformationButton = new
System.Windows.Forms.PictureBox();
        this.TragedyRepresentationOnCeramicImage = new
System.Windows.Forms.PictureBox();
        this.SuspendLayout();
        //
        // mainMenu1
        //
        this.mainMenu1.MenuItems.Add(this.menuItem1);
        this.mainMenu1.MenuItems.Add(this.BackButton);
        //
        // menuItem1
        //
        this.menuItem1.MenuItems.Add(this.TragedyWritersMenuButton);

this.menuItem1.MenuItems.Add(this.AncientGreekTheaterMenuButton);
        this.menuItem1.MenuItems.Add(this.BasicInformationMenuButton);
        this.menuItem1.Text = "Menu";
        //
        // TragedyWritersMenuButton
        //
        this.TragedyWritersMenuButton.Text = "Tragedy Writers";
        this.TragedyWritersMenuButton.Click += new
System.EventHandler(this.TragedyWritersMenuButton_Click);
        //
        // AncientGreekTheaterMenuButton
        //
        this.AncientGreekTheaterMenuButton.Text = "Ancient Greek
Theater";
        this.AncientGreekTheaterMenuButton.Click += new
System.EventHandler(this.AncientGreekTheaterMenuButton_Click);
        //
        // BasicInformationMenuButton
        //
        this.BasicInformationMenuButton.Text = "Basic Information";
        this.BasicInformationMenuButton.Click += new
System.EventHandler(this.BasicInformationMenuButton_Click);
        //
        // BackButton
        //
        this.BackButton.Text = "Back";
        this.BackButton.Click += new
System.EventHandler(this.BackButton_Click);
        //
        // TragedyWritersButton
        //
        this.TragedyWritersButton.Image =
((System.Drawing.Image)(resources.GetObject("TragedyWritersButton.Image"))
);
        this.TragedyWritersButton.Location = new
System.Drawing.Point(83, 155);
        this.TragedyWritersButton.Name = "TragedyWritersButton";
        this.TragedyWritersButton.Size = new System.Drawing.Size(70,
66);
        this.TragedyWritersButton.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        this.TragedyWritersButton.Tag = "Tragedy writers button";
        this.TragedyWritersButton.Click += new
System.EventHandler(this.TragedyWritersButton_Click);
        //

```

```

        // AncientGreekTheaterButton
        //
        this.AncientGreekTheaterButton.Image =
((System.Drawing.Image)(resources.GetObject("AncientGreekTheaterButton.Ima
ge")));
        this.AncientGreekTheaterButton.Location = new
System.Drawing.Point(149, 47);
        this.AncientGreekTheaterButton.Name =
"AncientGreekTheaterButton";
        this.AncientGreekTheaterButton.Size = new
System.Drawing.Size(70, 66);
        this.AncientGreekTheaterButton.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        this.AncientGreekTheaterButton.Tag = "Ancient greek theater
button";
        this.AncientGreekTheaterButton.Click += new
System.EventHandler(this.AncientGreekTheaterButton_Click);
        //
        // BasicInformationButton
        //
        this.BasicInformationButton.Image =
((System.Drawing.Image)(resources.GetObject("BasicInformationButton.Image"
))));
        this.BasicInformationButton.Location = new
System.Drawing.Point(20, 47);
        this.BasicInformationButton.Name = "BasicInformationButton";
        this.BasicInformationButton.Size = new System.Drawing.Size(70,
66);
        this.BasicInformationButton.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        this.BasicInformationButton.Tag = "Basic information button";
        this.BasicInformationButton.Click += new
System.EventHandler(this.BasicInformationButton_Click);
        //
        // TragedyRepresentationOnCeramicImage
        //
        this.TragedyRepresentationOnCeramicImage.Image =
((System.Drawing.Image)(resources.GetObject("TragedyRepresentationOnCeramic
Image.Image")));
        this.TragedyRepresentationOnCeramicImage.Location = new
System.Drawing.Point(0, 0);
        this.TragedyRepresentationOnCeramicImage.Name =
"TragedyRepresentationOnCeramicImage";
        this.TragedyRepresentationOnCeramicImage.Size = new
System.Drawing.Size(240, 268);
        this.TragedyRepresentationOnCeramicImage.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        this.TragedyRepresentationOnCeramicImage.Tag = "Tragedy
representation on ceramic";
        //
        // ChoicesPage
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(96F, 96F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Dpi;
        this.AutoScroll = true;
        this.ClientSize = new System.Drawing.Size(240, 268);
        this.Controls.Add(this.TragedyWritersButton);
        this.Controls.Add(this.AncientGreekTheaterButton);
        this.Controls.Add(this.BasicInformationButton);
        this.Controls.Add(this.TragedyRepresentationOnCeramicImage);
        this.Menu = this.mainMenu1;
        this.Name = "ChoicesPage";

```

```

        this.Tag = "Choices Page";
        this.Text = "Choices Page";
        this.ResumeLayout(false);
    }

    #endregion

    private System.Windows.Forms.PictureBox
TragedyRepresentationOnCeramicImage;
    private System.Windows.Forms.PictureBox BasicInformationButton;
    private System.Windows.Forms.PictureBox AncientGreekTheaterButton;
    private System.Windows.Forms.PictureBox TragedyWritersButton;
    private System.Windows.Forms.MenuItem menuItem1;
    private System.Windows.Forms.MenuItem TragedyWritersMenuButton;
    private System.Windows.Forms.MenuItem AncientGreekTheaterMenuButton;
    private System.Windows.Forms.MenuItem BasicInformationMenuButton;
    private System.Windows.Forms.MenuItem BackButton;
}
}

```

## Form3 – Basic Information:

### Κώδικας ενεργειών:

```

using System;
using System.Linq;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Media;
using System.IO;

namespace Diploma_Work
{
    public partial class BasicInformation : Form
    {
        SoundPlayer Media = new SoundPlayer(); /*Κατασκευή αντικειμένου
*προγράμματος αναπαραγωγής ήχου με το όνομα Media
*/

        public BasicInformation()
        {
            InitializeComponent();
        }

        /* Δημιουργία μεθόδου μη επιστρεφόμενης τιμής και χωρίς
* παραμέτρους για την επαναχρησιμοποίηση κώδικα
* στις επόμενες μεθόδους των events των αντικειμένων
* που χρησιμοποιούν τον ίδιο (κώδικα) με τον παρακάτω
*/
        void ObjectsVisibilityFalse()
        {

```

```

        HideTheTextButton.Visible = false;    /*Απόκρυψη του Button
*HideTheTextButton
*/
        increaseTextButton.Visible = false;    /*Απόκρυψη του Button
*increaseTextButton
*/
        decreaseTextButton.Visible = false;    /*Απόκρυψη του Button
*decreaseTextButton
*/
        PlayButton.Visible = false;    //Απόκρυψη του button PlayButton
        StopButton.Visible = false;    //Απόκρυψη του button StopButton
        HearTheTextLabel.Visible = false;    /*Απόκρυψη του label
*HearTheTextLabel
*/
    }

    void SoundObjectsVisibilityFalse()
    {
        PlayButton.Visible = false;
        StopButton.Visible = false;
        HearTheTextLabel.Visible = false;
    }

    private void BasicInformation_Activated(object sender, EventArgs
e) /*Μέθοδος για το event Activate (ενεργοποίηση) της φόρμας
*BasicInformation
*/
    {
        ObjectsVisibilityFalse();    /*Κλήση μεθόδου με το όνομα
*ObjectsVisibilityFalse
*/
    }

    private void BackButton_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem BackButton
    {
        ChoicesPage ChoicesPage = new ChoicesPage();    /*Κατασκευή
*αντικειμένου ChoicesPage με εμβέλεια τοπική (μεθόδου)
*/
        ChoicesPage.Show();    //Εμφάνιση της φόρμας ChoicesPage
        this.Hide();    //Απόκρυψη της παρούσας φόρμας
    }

    /* Δημιουργία μεθόδου μη επιστρεφόμενης τιμής και χωρίς
    * παραμέτρους για την επαναχρησιμοποίηση κώδικα
    * στις επόμενες μεθόδους των events των αντικειμένων
    * που χρησιμοποιούν τον ίδιο (κώδικα) με τον παρακάτω
    */
    void ObjectsVisibilityTrue()
    {
        HideTheTextButton.Visible = true;    /*Εμφάνιση του Button
*HideTheTextButton
*/
        increaseTextButton.Visible = true;    /*Εμφάνιση του Button
*increaseTextButton
*/
        decreaseTextButton.Visible = true;    /*Εμφάνιση του Button
*decreaseTextButton
*/
    }

```



```

        private void TragedyText_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem TragedyText
    {
        BasicInformationText.Visible = true;    /*Εμφάνιση του TextBox
*BasicInformationText
*/
        BasicInformationText.Text = "Tragedy is a form of art based on
human suffering that offers its audience pleasure. While most cultures
have developed forms that provoke this paradoxical response, tragedy
refers to a specific tradition of drama that has played a unique and
important role historically in the self-definition of Western
civilization. That tradition has been multiple and discontinuous, yet the
term has often been used to invoke a powerful effect of cultural identity
and historical continuity – \"the Greeks and the Elizabethans, in one
cultural form; Hellens and Christians, in a common activity,\" as Williams
puts it. From its obscure origins in the theaters of Athens 2500 years
ago, from which there survives only a fraction of the work of Aeschylus,
Sophocles and Euripides, through its singular articulations in the works
of Shakespeare, Lope de Vega, Racine or Schiller, to the more recent
naturalistic tragedy of Strindberg, Beckett's modernist meditations on
death, loss and suffering, or Müller's postmodernist reworkings of the
tragic canon, tragedy has remained an important site of cultural
experimentation, negotiation, struggle, and change. A long line of
philosophers – which includes Plato, Aristotle, Saint Augustine, Voltaire,
Hume, Diderot, Hegel, Schopenhauer, Kierkegaard, Nietzsche, Freud,
Benjamin, Camus, Lacan, and Deleuze–have analysed, speculated upon and
criticised the tragic form. In the wake of Aristotle's Poetics (335 BCE),
tragedy has been used to make genre distinctions, whether at the scale of
poetry in general, where the tragic divides against epic and lyric, or at
the scale of the drama, where tragedy is opposed to comedy. In the modern
era, tragedy has also been defined against drama, melodrama, the
tragicomic and epic theatre. ";    /*Αλλαγή κειμένου στο TextBox
*BasicInformationText
*/
        BasicInformationText.Tag = "Basic Information Text - Tragedy";
//Αλλαγή του Tag του TextBox BasicInformationText
        TextTitleLabel.Font = new Font("Tahoma", 13, FontStyle.Bold);
/*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
*TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
*γραμματοσειράς, στυλ γραμματοσειράς
*/
        TextTitleLabel.Text = "Tragedy";    /*Αλλαγή κειμένου στο Label
*TextTitleLabel
*/
        TextTitleLabel.Tag = "Text Title Label - Tragedy";    /*Αλλαγή
*του Tag στο Label TextTitleLabel
*/
        PlayButton.Visible = true;    //Εμφάνιση του button PlayButton
        HearTheTextLabel.Visible = true;    /*Εμφάνιση του label
*HearTheTextLabel
*/
        ObjectsVisibilityTrue();    /*Κλήση μεθόδου με το όνομα
*ObjectsVisibilityTrue
*/
    }

    private void OriginOfTragedyText_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem OriginOfTragedyText
    {
        BasicInformationText.Visible = true;    /*Εμφάνιση του TextBox
BasicInformationText
*/

```

```

        BasicInformationText.Text = "The word \"tragedy\" appears to
have been used to describe different phenomena at different times. It
derives from (Classical Greek τραγῳδία), contracted from trag(o)-aoidiā =
\"goat song\", which comes from tragos = \"goat\" and aeidein = \"to
sing\". Scholars suspect this may be traced to a time when a goat was
either the prize in a competition of choral dancing or was that around
which a chorus danced prior to the animal's ritual sacrifice. Writing in
335 BCE (long after the Golden Age of 5th-century Athenian tragedy),
Aristotle provides the earliest-surviving explanation for the origin of
the dramatic art-form in his Poetics, in which he argues that tragedy
developed from the improvisations of the leader of choral dithyrambs
(hymns sung and danced in praise of Dionysos, the god of wine and
fertility): At any rate it originated in improvisation—both tragedy itself
and comedy. The one tragedy came from the prelude to the dithyramb and the
other comedy from the prelude to the phallic songs which still survive as
institutions in many cities. Tragedy then gradually evolved as men
developed each element that came to light and after going through many
changes, it stopped when it had found its own natural form (IV, 1449a).
There is some dissent to the dithyrambic origins of tragedy mostly based
in the differences between the shapes of their choruses and styles of
dancing. A common descent from pre-Hellenic fertility and burial rites has
been suggested. Nietzsche discussed the origins of Greek tragedy in his
early book, The Birth of Tragedy (1872)."; /*Αλλαγή κειμένου στο TextBox
*BasicInformationText
*/
        TextTitleLabel.Tag = "Text Title Label - Origin of Tragedy";
//Αλλαγή του Tag στο Label TextTitleLabel
        TextTitleLabel.Font = new Font("Tahoma", 10, FontStyle.Bold);
/*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
*TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
*γραμματοσειράς, στυλ γραμματοσειράς*/
        TextTitleLabel.Tag = "Text Title Label - Origin of Tragedy";
//Αλλαγή του Tag στο Label TextTitleLabel
        TextTitleLabel.Text = "Origin of Tragedy"; /*Αλλαγή κειμένου
*στο Label TextTitleLabel
*/
        ObjectsVisibilityTrue(); /*Κλήση μεθόδου με το όνομα
*ObjectsVisibilityTrue
*/
        SoundObjectsVisibilityFalse();
    }

    private void PerformanceOfGreekTragediesText_Click(object sender,
EventArgs e) /*Μέθοδος για το event Click του MenuItem
*PerformanceOfGreekTragediesText
*/
    {
        BasicInformationText.Visible = true; /*Εμφάνιση του TextBox
*BasicInformationText
*/
        BasicInformationText.Text = "Greek literature boasts three
great writers of tragedy whose works are extant: Aeschylus, Sophocles, and
Euripides. The largest festival for Greek tragedy was the Dionysia held
for five days in March, for which competition prominent tragedians usually
submitted three tragedies and one satyr play each. Greek tragedies were
performed in late March/early April at an annual state religious festival
in honor of Dionysus. The presentation took the form of a contest among
three playwrights, who presented their works on three successive days.
Each playwright would prepare a trilogy of tragedies, plus an unrelated
concluding comic piece called a satyr play. Often, the three plays
featured linked stories, but later writers like Euripides may have
presented three unrelated plays. Only one complete trilogy has survived,

```

the Oresteia of Aeschylus. The Greek theatre was in the open air, on the side of a hill, and performances of a trilogy and satyr play probably lasted most of the day. Performances were apparently open to all citizens, including women, but evidence is scanty. The theatre of Dionysus at Athens probably held around 12,000 people. All of the choral parts were sung (to flute accompaniment) and some of the actors' answers to the chorus were sung as well. The play as a whole was composed in various verse meters. All actors were male and wore masks. A Greek chorus danced as well as sang, though no one knows exactly what sorts of steps the chorus performed as it sang. Choral songs in tragedy are often divided into three sections: strophe ("turning, circling"), antistrophe ("counter-turning, counter-circling") and epode ("after-song"). Many ancient Greek tragedians employed the ekkyklêma as a theatrical device, which was a cart hidden behind the scenery which could be rolled out to display the aftermath of some event which had happened out of sight of the audience. This event was frequently a brutal murder of some sort, an act of violence which could not be effectively portrayed visually, but an action of which the other characters must see the effects in order for it to have meaning and emotional resonance. Another reason that the violence happened off stage was that the theatre was considered a holy place, so to kill someone on stage is to kill them in the real world. A prime example of the use of the ekkyklêma is after the murder of Agamemnon in the first play of Aeschylus' Oresteia, when the king's butchered body is wheeled out in a grand display for all to see. Variations on the ekkyklêma are used in tragedies and other forms to this day, as writers still find it a useful and often powerful device for showing the consequences of extreme human actions. Another such device was a crane, the mechane, which served to hoist a god or goddess on stage when they were supposed to arrive flying. This device gave origin to the phrase "deus ex machina" ("god out of a machine"), that is, the surprise intervention of an unforeseen external factor that changes the outcome of an event. Greek tragedies also sometimes included a chorus composed of singers to advance and fill in detail of the plot. ";

```
//Αλλαγή κειμένου στο TextBox BasicInformationText
    TextTitleLabel.Tag = "Text Title Label - Performance of Greek
Tragedies"; //Αλλαγή του Tag στο Label TextTitleLabel
    TextTitleLabel.Font = new Font("Arial", 10, FontStyle.Bold);
/*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
*TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
*γραμματοσειράς, στυλ γραμματοσειράς */
    TextTitleLabel.Text = "Performance of Greek Tragedies";
//Αλλαγή κειμένου στο Label TextTitleLabel
    TextTitleLabel.Tag = "Text Title Label - Performance of Greek
Tragedies"; //Αλλαγή του Tag στο Label TextTitleLabel
    ObjectsVisibilityTrue(); /*Κλήση μεθόδου με το όνομα
*ObjectsVisibilityTrue
*/
    SoundObjectsVisibilityFalse();
}

private void ReadTheTextButton_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του Button ReadTheTextButton
{
    BasicInformationText.Visible = true; /*Εμφάνιση του κειμένου
*της φόρμας (του TextBox BasicInformationText)
*/
    ReadTheTextButton.Visible = false; /*Απόκρυψη του Button
*ReadTheTextButton
*/
    ObjectsVisibilityTrue(); /*Κλήση μεθόδου με το όνομα
*ObjectsVisibilityTrue
*/
    if (TextTitleLabel.Text == "Tragedy")
```

```

        {
            PlayButton.Visible = true;
            StopButton.Visible = true;
            HearTheTextLabel.Visible = true;
        }
    }

    private void HideTheTextButton_Click(object sender, EventArgs e)
    //Μέθοδος για το event Click του button HideTheTextButton
    {
        BasicInformationText.Visible = false; /*Απόκρυψη του TextBox
    *BasicInformationText
    */
        ReadTheTextButton.Visible = true; /*Εμφάνιση του Button
    *ReadTheTextButton
    */
        ObjectsVisibilityFalse(); /*Κλήση μεθόδου με το όνομα
    *ObjectsVisibilityFalse
    */
        SoundObjectsVisibilityFalse();
    }

    private void increaseTextButton_Click(object sender, EventArgs e)
    //Μέθοδος για το event Click του button increaseTextButton
    {
        float k = BasicInformationText.Font.Size; /*Δήλωση μεταβλητή
    *για το Size της Font του TextBox BasicInformationText, τύπου float
    */
        if (k <= 18) //Εντολή απόφασης για τον έλεγχο της τιμής του k
        {
            /* Κατασκευή αντικειμένου για το Font του TextBox
    *BasicInformationText.
            * Στη 2η παράμετρο (fontSize) γίνεται αύξηση κατά 1em στο
    *υπάρχον Size της Font του TextBox BasicInformationText
            */
            BasicInformationText.Font = new Font("Arial", k + 1,
    FontStyle.Regular);
        }
    }

    private void decreaseTextButton_Click(object sender, EventArgs e)
    {
        float k = BasicInformationText.Font.Size; //Δήλωση μεταβλητή
    *για το Size της Font του TextBox BasicInformationText, τύπου float
    */
        if (k >= 8) //Εντολή απόφασης για τον έλεγχο της τιμής του k
        {
            /* Κατασκευή αντικειμένου για το Font του TextBox
    *BasicInformationText.
            * Στη 2η παράμετρο (fontSize) γίνεται μείωση κατά 1em στο
    *υπάρχον Size της Font του TextBox BasicInformationText
            */
            BasicInformationText.Font = new Font("Arial", k - 1,
    FontStyle.Regular);
        }
    }

    private void PlayButton_Click(object sender, EventArgs e)
    //Κατασκευή μεθόδου για το κλικ του κουμπιού PlayButton
    {
        PlayButton.Visible = false; //Απόκρυψη του button PlayButton
        StopButton.Visible = true; //Εμφάνιση του button StopButton
    }

```

```

        Stream STR = new MemoryStream(Properties.Resources.Tragedy);
//Κατασκευαστής ρόης ήχου με το όνομα STR και άντληση αυτού από το source
        Media = new SoundPlayer(STR); /*Κατασκευστής προγράμματος
*αναπαραγωγής ήχου και ανάθεση αυτού στην τιμή Media
*/
        Media.Play(); //Αναπαραγωγή του ήχου
    }

    private void StopButton_Click(object sender, EventArgs e)
//Κατασκευή μεθόδου για το κλικ του κουμπιού StopButton
    {
        PlayButton.Visible = true; //Εμφάνιση του button PlayButton
        StopButton.Visible = false; //Απόκρυψη του button StopButton
        Stream STR = new MemoryStream(Properties.Resources.Tragedy);
//Κατασκευαστής ρόης ήχου με το όνομα STR και άντληση αυτού από το source
        Media = new SoundPlayer(STR); /*Κατασκευστής προγράμματος
*αναπαραγωγής ήχου και ανάθεση αυτού στην τιμή Media
*/
        Media.Stop(); //Διακοπή αναπαραγωγής του ήχου
    }
}
}
}

```

## Κώδικας σχεδιασμού:

```

namespace Diploma_Work
{
    partial class BasicInformation
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
        disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
            System.ComponentModel.ComponentResourceManager(typeof(BasicInformation));
            this.mainMenu1 = new System.Windows.Forms.MainMenu();
            this.OtherInfosMenu = new System.Windows.Forms.MenuItem();

```

```

        this.PerformanceOfGreekTragediesText = new
System.Windows.Forms.MenuItem();
        this.OriginOfTragedyText = new
System.Windows.Forms.MenuItem();
        this.TragedyText = new System.Windows.Forms.MenuItem();
        this.BackButton = new System.Windows.Forms.MenuItem();
        this.BasicInformationText = new
System.Windows.Forms.TextBox();
        this.TextTitleLabel = new System.Windows.Forms.Label();
        this.ReadTheTextButton = new System.Windows.Forms.Button();
        this.HideTheTextButton = new System.Windows.Forms.Button();
        this.increaseTextButton = new System.Windows.Forms.Button();
        this.decreaseTextButton = new System.Windows.Forms.Button();
        this.HearTheTextLabel = new System.Windows.Forms.Label();
        this.PlayButton = new System.Windows.Forms.PictureBox();
        this.SatyrImage = new System.Windows.Forms.PictureBox();
        this.StopButton = new System.Windows.Forms.PictureBox();
        this.SuspendLayout();
        //
        // mainMenu1
        //
        this.mainMenu1.MenuItems.Add(this.OtherInfosMenu);
        this.mainMenu1.MenuItems.Add(this.BackButton);
        //
        // OtherInfosMenu
        //
        this.OtherInfosMenu.MenuItems.Add(this.PerformanceOfGreekTragediesText);
        this.OtherInfosMenu.MenuItems.Add(this.OriginOfTragedyText);
        this.OtherInfosMenu.MenuItems.Add(this.TragedyText);
        this.OtherInfosMenu.Text = "Other Infos";
        //
        // PerformanceOfGreekTragediesText
        //
        this.PerformanceOfGreekTragediesText.Text = "Performance of
Greek Tragedies";
        this.PerformanceOfGreekTragediesText.Click += new
System.EventHandler(this.PerformanceOfGreekTragediesText_Click);
        //
        // OriginOfTragedyText
        //
        this.OriginOfTragedyText.Text = "Origin of Tragedy";
        this.OriginOfTragedyText.Click += new
System.EventHandler(this.OriginOfTragedyText_Click);
        //
        // TragedyText
        //
        this.TragedyText.Text = "Tragedy";
        this.TragedyText.Click += new
System.EventHandler(this.TragedyText_Click);
        //
        // BackButton
        //
        this.BackButton.Text = "Back";
        this.BackButton.Click += new
System.EventHandler(this.BackButton_Click);
        //
        // BasicInformationText
        //
        this.BasicInformationText.BackColor =
System.Drawing.Color.White;

```

```

        this.BasicInformationText.Location = new
System.Drawing.Point(9, 20);
        this.BasicInformationText.Multiline = true;
        this.BasicInformationText.Name = "BasicInformationText";
        this.BasicInformationText.ReadOnly = true;
        this.BasicInformationText.ScrollBars =
System.Windows.Forms.ScrollBars.Vertical;
        this.BasicInformationText.Size = new System.Drawing.Size(222,
170);

        this.BasicInformationText.TabIndex = 1;
        this.BasicInformationText.Tag = "Basic Information Text";
        this.BasicInformationText.Text =
resources.GetString("BasicInformationText.Text");
        this.BasicInformationText.Visible = false;
        //
        // TextTitleLabel
        //
        this.TextTitleLabel.BackColor =
System.Drawing.Color.Transparent;
        this.TextTitleLabel.Font = new System.Drawing.Font("Arial",
13F, System.Drawing.FontStyle.Bold);
        this.TextTitleLabel.Location = new System.Drawing.Point(0, 0);
        this.TextTitleLabel.Name = "TextTitleLabel";
        this.TextTitleLabel.Size = new System.Drawing.Size(240, 20);
        this.TextTitleLabel.Tag = "Text title label";
        this.TextTitleLabel.Text = "Tragedy";
        this.TextTitleLabel.TextAlign =
System.Drawing.ContentAlignment.TopCenter;
        //
        // ReadTheTextButton
        //
        this.ReadTheTextButton.Location = new System.Drawing.Point(9,
244);

        this.ReadTheTextButton.Name = "ReadTheTextButton";
        this.ReadTheTextButton.Size = new System.Drawing.Size(222,
20);

        this.ReadTheTextButton.TabIndex = 3;
        this.ReadTheTextButton.Tag = "Read the text button";
        this.ReadTheTextButton.Text = "Read the Text";
        this.ReadTheTextButton.Click += new
System.EventHandler(this.ReadTheTextButton_Click);
        //
        // HideTheTextButton
        //
        this.HideTheTextButton.Location = new System.Drawing.Point(9,
244);

        this.HideTheTextButton.Name = "HideTheTextButton";
        this.HideTheTextButton.Size = new System.Drawing.Size(222,
20);

        this.HideTheTextButton.TabIndex = 6;
        this.HideTheTextButton.Tag = "Hide the text button";
        this.HideTheTextButton.Text = "Hide the Text";
        this.HideTheTextButton.Click += new
System.EventHandler(this.HideTheTextButton_Click);
        //
        // increaseTextButton
        //
        this.increaseTextButton.BackColor =
System.Drawing.SystemColors.GrayText;
        this.increaseTextButton.Font = new
System.Drawing.Font("Tahoma", 10F, System.Drawing.FontStyle.Bold);

```



```

        this.increaseTextButton.ForeColor =
System.Drawing.Color.Black;
        this.increaseTextButton.Location = new
System.Drawing.Point(170, 196);
        this.increaseTextButton.Name = "increaseTextButton";
        this.increaseTextButton.Size = new System.Drawing.Size(28,
20);

        this.increaseTextButton.TabIndex = 9;
        this.increaseTextButton.Tag = "Button for increasing the text
size";

        this.increaseTextButton.Text = "+A";
        this.increaseTextButton.Click += new
System.EventHandler(this.increaseTextButton_Click);
        //
        // decreaseTextButton
        //
        this.decreaseTextButton.BackColor =
System.Drawing.SystemColors.GrayText;
        this.decreaseTextButton.Font = new
System.Drawing.Font("Tahoma", 10F, System.Drawing.FontStyle.Bold);
        this.decreaseTextButton.ForeColor =
System.Drawing.Color.Black;
        this.decreaseTextButton.Location = new
System.Drawing.Point(203, 196);
        this.decreaseTextButton.Name = "decreaseTextButton";
        this.decreaseTextButton.Size = new System.Drawing.Size(28,
20);

        this.decreaseTextButton.TabIndex = 10;
        this.decreaseTextButton.Tag = "Button for decreasing the text
size";

        this.decreaseTextButton.Text = "-A";
        this.decreaseTextButton.Click += new
System.EventHandler(this.decreaseTextButton_Click);
        //
        // HearTheTextLabel
        //
        this.HearTheTextLabel.Font = new System.Drawing.Font("Tahoma",
10F, System.Drawing.FontStyle.Bold);
        this.HearTheTextLabel.Location = new System.Drawing.Point(92,
221);

        this.HearTheTextLabel.Name = "HearTheTextLabel";
        this.HearTheTextLabel.Size = new System.Drawing.Size(106, 17);
        this.HearTheTextLabel.Tag = "Hear the Text Label";
        this.HearTheTextLabel.Text = "Hear the Text:";
        this.HearTheTextLabel.TextAlign =
System.Drawing.ContentAlignment.TopRight;
        //
        // PlayButton
        //
        this.PlayButton.Image =
((System.Drawing.Image)(resources.GetObject("PlayButton.Image")));
        this.PlayButton.Location = new System.Drawing.Point(203, 220);
        this.PlayButton.Name = "PlayButton";
        this.PlayButton.Size = new System.Drawing.Size(28, 21);
        this.PlayButton.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        this.PlayButton.Tag = "Play the text reproduction";
        this.PlayButton.Click += new
System.EventHandler(this.PlayButton_Click);
        //
        // SatyrImage
        //

```

```

        this.SatyrImage.Image =
((System.Drawing.Image)(resources.GetObject("SatyrImage.Image")));
        this.SatyrImage.Location = new System.Drawing.Point(0, 0);
        this.SatyrImage.Name = "SatyrImage";
        this.SatyrImage.Size = new System.Drawing.Size(240, 268);
        this.SatyrImage.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        this.SatyrImage.Tag = "Satyr";
        //
        // StopButton
        //
        this.StopButton.Image =
((System.Drawing.Image)(resources.GetObject("StopButton.Image")));
        this.StopButton.Location = new System.Drawing.Point(203, 220);
        this.StopButton.Name = "StopButton";
        this.StopButton.Size = new System.Drawing.Size(28, 21);
        this.StopButton.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        this.StopButton.Tag = "Stop the text reproduction";
        this.StopButton.Click += new
System.EventHandler(this.StopButton_Click);
        //
        // BasicInformation
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(96F, 96F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Dpi;
        this.AutoScroll = true;
        this.ClientSize = new System.Drawing.Size(240, 268);
        this.Controls.Add(this.PlayButton);
        this.Controls.Add(this.HearTheTextLabel);
        this.Controls.Add(this.decreaseTextButton);
        this.Controls.Add(this.increaseTextButton);
        this.Controls.Add(this.HideTheTextButton);
        this.Controls.Add(this.ReadTheTextButton);
        this.Controls.Add(this.TextTitleLabel);
        this.Controls.Add(this.BasicInformationText);
        this.Controls.Add(this.StopButton);
        this.Controls.Add(this.SatyrImage);
        this.KeyPreview = true;
        this.Menu = this.mainMenu1;
        this.Name = "BasicInformation";
        this.Tag = "Basic information";
        this.Text = "Basic Information";
        this.Activated += new
System.EventHandler(this.BasicInformation_Activated);
        this.ResumeLayout(false);

    }

#endregion

private System.Windows.Forms.PictureBox SatyrImage;
private System.Windows.Forms.MainMenu mainMenu1;
private System.Windows.Forms.TextBox BasicInformationText;
private System.Windows.Forms.Label TextTitleLabel;
private System.Windows.Forms.MenuItem OtherInfosMenu;
private System.Windows.Forms.MenuItem
PerformanceOfGreekTragediesText;
private System.Windows.Forms.MenuItem BackButton;
private System.Windows.Forms.MenuItem OriginOfTragedyText;
private System.Windows.Forms.MenuItem TragedyText;
private System.Windows.Forms.Button ReadTheTextButton;

```

```

        private System.Windows.Forms.Button HideTheTextButton;
        private System.Windows.Forms.Button increaseTextButton;
        private System.Windows.Forms.Button decreaseTextButton;
        private System.Windows.Forms.Label HearTheTextLabel;
        private System.Windows.Forms.PictureBox PlayButton;
        private System.Windows.Forms.PictureBox StopButton;
    }
}

```

## Form4 – Ancient Greek Theater:

### Κώδικας ενεργειών:

```

using System;
using System.Linq;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Diploma_Work
{
    public partial class AncientGreekTheater : Form
    {
        public AncientGreekTheater()
        {
            InitializeComponent();
        }

        private void AncientGreekTheater_Activated(object sender,
EventArgs e) /*Μέθοδος για το event Activated (ενεργοποιημένο) της
*φόρμας AncientGreekTheater*/
        {
            HideTheTextButton.Visible = false; /*Απόκρυψη του button
*HideTheTextButton*/
            FirstLinkLabel.Visible = false; /*Απόκρυψη του LinkLabel
*FirstLinkLabel*/
            SecondLinkLabel.Visible = false; /*Απόκρυψη του LinkLabel
*SecondLinkLabel*/
            increaseTextButton.Visible = false; /*Απόκρυψη του button
*increaseTextButton*/
            decreaseTextButton.Visible = false; /*Απόκρυψη του button
*decreaseTextButton*/
        }

        private void BackButton_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem BackButton
        {
            ChoicesPage ChoicesPage = new ChoicesPage(); /*Κατασκευή
*αντικειμένου ChoicesPage με εμβέλεια τοπική (μεθόδου) */
            ChoicesPage.Show(); //Εμφάνιση της φόρμας ChoicesPage
            this.Hide(); //Αποκρυψη της παρούσας φόρμας
        }
    }
}

```

```

        private void ReadTheTextButton_Click(object sender, EventArgs e)
//Μέθοδος για το Event Click του button ReadTheTextButton
    {
        AncientGreekTheaterText.Visible = true; /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
        ReadTheTextButton.Visible = false; /*Απόκρυψη του Button
*ReadTheTextButton*/
        HideTheTextButton.Visible = true; /*Εμφάνιση του Button
*HideTheTextButton*/
        increaseTextButton.Visible = true; /*Εμφάνιση του button
*increaseTextButton*/
        decreaseTextButton.Visible = true; /*Εμφάνιση του button
*decreaseTextButton */
        /*
        * Χρήση εντολής if για τη λήψη απόφασης
        * για τον έλεγχο του Text του Label TextTitleLabel (έλεγχος
για το αν ισχύει μία από τις τρεις αναγραφόμενες περιπτώσεις)
        * και για το visibility του TextBox AncientGreekTheaterText
(έλεγχος για το αν αυτό θα είναι εμφανές)
        * Για να μπει το πρόγραμμα στον έλεγχο πρέπει να ισχύουν και
τα δύο παραπάνω ώστε να εκτελεστούν οι ενέργειες που βρίσκονται μέσα σε
αυτόν
        * και αφορούν το visibility των LinkLabels που υπάρχουν στη
φόρμα
        */
        if (((TextTitleLabel.Text == "Audience") ||
(TextTitleLabel.Text == "Skene") || (TextTitleLabel.Text == "Orchestra"))
&& (AncientGreekTheaterText.Visible == true))
        {
            FirstLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*FirstLinkLabel*/
            SecondLinkLabel.Visible = true; /*Απόκρυψη του LinkLabel
*SecondLinkLabel*/
        }
    }

    private void AncientGreekTheaterMenuItem_Click(object sender,
EventArgs e) /*Μέθοδος για το event Click του MenuItem
*AncientGreekTheaterMainText*/
    {
        AncientGreekTheaterText.Text = "The Greek theatre history
began with festivals honoring their gods. A god, Dionysus, was honored
with a festival called by \"City Dionysia\". In Athens, during this
festival, men used to perform songs to welcome Dionysus. Plays were only
presented at City Dionysia festival. Athens was the main center for these
theatrical traditions. Athenians spread these festivals to its numerous
allies in order to promote a common identity. At the early Greek
festivals, the actors, directors, and dramatists were all the same person.
After some time, only three actors were allowed to perform in each play.
Later few non-speaking roles were allowed to perform on-stage. Due to
limited number of actors allowed on-stage, the chorus evolved into a very
active part of Greek theatre. Music was often played during the chorus'
delivery of its lines. Tragedy, comedy, and satyr plays were the
theatrical forms. Tragedy and comedy were viewed as completely separate
genres. Satyr plays dealt with the mythological subject in comic manner.
Aristotle's Poetics sets out a thesis about the perfect structure for
tragedy. "; //Αλλαγή κειμένου στο TextBox AncientGreekTheaterText
        AncientGreekTheaterText.Tag = "Ancient Greek Theater Text -
Greek Theater"; //Αλλαγή του Tag του TextBox AncientGreekTheaterText
        TextTitleLabel.Font = new Font("Arial", 13, FontStyle.Bold);
/*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label

```

```

*TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
*γραμματοσειράς, στυλ γραμματοσειράς*/
    TextTitleLabel.Text = "Ancient Greek Theater"; /*Αλλαγή
*κειμένου στο Label TextTitleLabel*/
    TextTitleLabel.Tag = "Text Title Label - Ancient Greek
Theater"; //Αλλαγή του Tag στο Label TextTitleLabel
    increaseTextButton.Visible = true; /*Εμφάνιση του button
*increaseTextButton*/
    decreaseTextButton.Visible = true; /*Εμφάνιση του button
*decreaseTextButton*/
    FirstLinkLabel.Visible = false; /*Απόκρυψη του LinkLabel
*SecondLinkLabel*/
    SecondLinkLabel.Visible = false; /*Απόκρυψη του LinkLabel
*SecondLinkLabel*/
    }

    private void TheaterBuildingsMenuItem_Click(object sender,
EventArgs e)
    {
        AncientGreekTheaterText.Visible = true; /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
        AncientGreekTheaterText.Text = "Theater buildings were called
a theatron. The theaters were large, open-air structures constructed on
the slopes of hills. They consisted of three main elements: the orchestra,
the skene, and the audience. Orchestra: A large circular or rectangular
area at the center part of the theatre, where the play, dance, religious
rites, acting used to take place. Skene: A large rectangular building
situated behind the orchestra, used as a backstage. Actors could change
their costumes and masks. Earlier the skene was a tent or hut, later it
became a permanent stone structure. These structures were sometimes
painted to serve as backdrops. Audience: Rising from the circle of the
orchestra was the audience. The theatres were originally built on a very
large scale to accommodate the large number of people on stage, as well as
the large number of people in the audience, up to fourteen thousand.";
//Αλλαγή κειμένου στο TextBox AncientGreekTheaterText
        AncientGreekTheaterText.Tag = "Ancient Greek Theater Text -
Theater Buildings"; //Αλλαγή του Tag του TextBox AncientGreekTheaterText
        TextTitleLabel.Font = new Font("Arial", 11, FontStyle.Bold);
/*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
*TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
*γραμματοσειράς, στυλ γραμματοσειράς*/
        TextTitleLabel.Text = "Theater Buildings"; /*Αλλαγή κειμένου
στο Label TextTitleLabel
        TextTitleLabel.Tag = "Text Title Label - Theater Buildings";
//Αλλαγή του Tag στο Label TextTitleLabel*/
        ReadTheTextButton.Visible = false; /*Απόκρυψη του Button
*ReadTheTextButton*/
        HideTheTextButton.Visible = true; /*Εμφάνιση του Button
*HideTheTextButton*/
        FirstLinkLabel.Visible = false; /*Απόκρυψη του LinkLabel
*SecondLinkLabel*/
        SecondLinkLabel.Visible = false; /*Απόκρυψη του LinkLabel
*SecondLinkLabel*/
        increaseTextButton.Visible = true; /*Εμφάνιση του button
*increaseTextButton*/
        decreaseTextButton.Visible = true; /*Εμφάνιση του button
*decreaseTextButton*/
    }

    private void AudienceTextMenuItem_Click(object sender, EventArgs
e)
    {

```

```

        AncientGreekTheaterText.Visible = true; /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
        AncientGreekTheaterText.Text = "Rising from the circle of the
orchestra was the audience. The theatres were originally built on a very
large scale to accommodate the large number of people on stage, as well as
the large number of people in the audience, up to fourteen thousand.";
//Αλλαγή κειμένου στο TextBox AncientGreekTheaterText
        AncientGreekTheaterText.Tag = "Ancient Greek Theater Text -
Audience"; //Αλλαγή του Tag του TextBox AncientGreekTheaterText
        TextTitleLabel.Font = new Font("Arial", 10, FontStyle.Bold);
/*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
*TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
*γραμματοσειράς, στυλ γραμματοσειράς*/
        TextTitleLabel.Text = "Audience"; /*Αλλαγή κειμένου στο
*Label TextTitleLabel*/
        TextTitleLabel.Tag = "Text Title Label - Audience"; /*Αλλαγή
*του Tag στο Label TextTitleLabel*/
        FirstLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*FirstLinkLabel*/
        FirstLinkLabel.Text = "Skene"; /*Αλλαγή Text στο LinkLabel
*FirstLinkLabel*/
        FirstLinkLabel.Tag = "Skene Text Link"; /*Αλλαγή του Tag στο
*LinkLabel FirstLinkLabel*/
        SecondLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*SecondLinkLabel*/
        SecondLinkLabel.Text = "Orchestra"; /*Αλλαγή Text στο
*LinkLabel SecondLinkLabel*/
        SecondLinkLabel.Tag = "Orchestra Text Link"; /*Αλλαγή του
*Tag στο LinkLabel SecondLinkLabel*/
        ReadTheTextButton.Visible = false; /*Απόκρυψη του Button
*ReadTheTextButton*/
        HideTheTextButton.Visible = true; /*Εμφάνιση του Button
*HideTheTextButton*/
        increaseTextButton.Visible = true; /*Εμφάνιση του button
*increaseTextButton*/
        decreaseTextButton.Visible = true; /*Εμφάνιση του button
*decreaseTextButton*/
    }

    private void SkeneTextMenuItem_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem SceneText
    {
        AncientGreekTheaterText.Visible = true; /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
        AncientGreekTheaterText.Text = "A large rectangular building
situated behind the orchestra, used as a backstage. Actors could change
their costumes and masks. Earlier the skene was a tent or hut, later it
became a permanent stone structure. These structures were sometimes
painted to serve as backdrops."; /*Αλλαγή κειμένου στο TextBox
*AncientGreekTheaterText*/
        AncientGreekTheaterText.Tag = "Ancient Greek Theater Text -
Skene"; //Αλλαγή του Tag του TextBox AncientGreekTheaterText
        TextTitleLabel.Font = new Font("Arial", 10, FontStyle.Bold);
/*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
*TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
*γραμματοσειράς, στυλ γραμματοσειράς*/
        TextTitleLabel.Text = "Skene"; /*Αλλαγή κειμένου στο Label
*TextTitleLabel*/
        TextTitleLabel.Tag = "Text Title Label - Skene"; /*Αλλαγή
*του Tag στο Label TextTitleLabel*/
        FirstLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*FirstLinkLabel*/

```

```

        FirstLinkLabel.Text = "Audience";    /*Αλλαγή Text στο
*LinkLabel FirstLinkLabel*/
        FirstLinkLabel.Tag = "Audience Text Link";    /*Αλλαγή του Tag
*στο LinkLabel FirstLinkLabel
        SecondLinkLabel.Visible = true;    /*Εμφάνιση του LinkLabel
*SecondLinkLabel*/
        SecondLinkLabel.Text = "Orchestra";    /*Αλλαγή Text στο
*LinkLabel SecondLinkLabel*/
        SecondLinkLabel.Tag = "Orchestra Text Link";    /*Αλλαγή του
*Tag στο LinkLabel SecondLinkLabel*/
        ReadTheTextButton.Visible = false;    /*Απόκρυψη του Button
*ReadTheTextButton*/
        HideTheTextButton.Visible = true;    /*Εμφάνιση του Button
*HideTheTextButton*/
        increaseTextButton.Visible = true;    /*Εμφάνιση του button
*increaseTextButton*/
        decreaseTextButton.Visible = true;    /*Εμφάνιση του button
*decreaseTextButton*/
    }

    private void OrchestraTextMenuItem_Click(object sender, EventArgs
e)    //Μέθοδος για το event Click του MenuItem OrchestraText
    {
        AncientGreekTheaterText.Visible = true;    /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
        AncientGreekTheaterText.Text = "A large circular or
rectangular area at the center part of the theatre, where the play, dance,
religious rites, acting used to take place.";    /*Αλλαγή κειμένου στο
*TextBox AncientGreekTheaterText*/
        AncientGreekTheaterText.Tag = "Ancient Greek Theater Text -
Orchestra";    //Αλλαγή του Tag του TextBox AncientGreekTheaterText
        TextTitleLabel.Font = new Font("Arial", 10, FontStyle.Bold);
/*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
*TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
*γραμματοσειράς, στυλ γραμματοσειράς*/
        TextTitleLabel.Text = "Orchestra";    /*Αλλαγή κειμένου στο
*Label TextTitleLabel*/
        TextTitleLabel.Tag = "Text Title Label - Orchestra";
//Αλλαγή του Tag στο Label TextTitleLabel
        FirstLinkLabel.Visible = true;    /*Εμφάνιση του LinkLabel
*FirstLinkLabel*/
        FirstLinkLabel.Text = "Audience";    /*Αλλαγή Text στο
*LinkLabel FirstLinkLabel*/
        FirstLinkLabel.Tag = "Audience Text Link";    /*Αλλαγή του Tag
*στο LinkLabel FirstLinkLabel*/
        SecondLinkLabel.Visible = true;    /*Εμφάνιση του LinkLabel
*SecondLinkLabel*/
        SecondLinkLabel.Text = "Skene";    /*Αλλαγή Text στο LinkLabel
*SecondLinkLabel*/
        SecondLinkLabel.Tag = "Skene Text Link";    /*Αλλαγή του Tag
*στο LinkLabel SecondLinkLabel*/
        ReadTheTextButton.Visible = false;    /*Απόκρυψη του Button
*ReadTheTextButton*/
        HideTheTextButton.Visible = true;    /*Εμφάνιση του Button
*HideTheTextButton*/
        increaseTextButton.Visible = true;    /*Εμφάνιση του button
*increaseTextButton*/
        decreaseTextButton.Visible = true;    /*Εμφάνιση του button
*decreaseTextButton*/
    }

```



```

        private void FirstLinkLabel_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του LinkLabel FirstLinkLabel
    {
        ReadTheTextButton.Visible = false; /*Απόκρυψη του Button
*ReadTheTextButton*/
        HideTheTextButton.Visible = true; /*Εμφάνιση του Button
*HideTheTextButton*/
        increaseTextButton.Visible = true; /*Εμφάνιση του button
*increaseTextButton*/
        decreaseTextButton.Visible = true; /*Εμφάνιση του button
*decreaseTextButton*/

        /*
        * Χρήση εντολής if για τη λήψη απόφασης.
        * Ο έλεγχος αφορά το Text του LinkLabel FirstLinkLabel
        * Ανάλογα με το Text, θα εμφανίζονται τα παρακάτω αντικείμενα
με συγκεκριμένες ιδιότητες
        * που αφορούν το συγκεκριμένο θέμα (του link)
        */
        if (FirstLinkLabel.Text == "Audience")
        {
            AncientGreekTheaterText.Visible = true; /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
            AncientGreekTheaterText.Text = "Rising from the circle of
the orchestra was the audience. The theatres were originally built on a
very large scale to accommodate the large number of people on stage, as
well as the large number of people in the audience, up to fourteen
thousand."; //Αλλαγή κειμένου στο TextBox AncientGreekTheaterText
            AncientGreekTheaterText.Tag = "Ancient Greek Theater Text
- Audience"; //Αλλαγή του Tag του TextBox AncientGreekTheaterText
            TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); /*Κατασκευή αντικείμενου Font για τη γραμματοσειρά του
*Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
*μέγεθος γραμματοσειράς, στυλ γραμματοσειράς*/
            TextTitleLabel.Text = "Audience"; /*Αλλαγή κειμένου στο
*Label TextTitleLabel*/
            TextTitleLabel.Tag = "Text Title Label - Audience";
//Αλλαγή του Tag στο Label TextTitleLabel
            FirstLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*FirstLinkLabel*/
            FirstLinkLabel.Text = "Skene"; /*Αλλαγή Text στο
*LinkLabel FirstLinkLabel*/
            FirstLinkLabel.Tag = "Skene Text Link"; /*Αλλαγή του Tag
*στο LinkLabel FirstLinkLabel*/
            SecondLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*SecondLinkLabel*/
            SecondLinkLabel.Text = "Orchestra"; /*Αλλαγή Text στο
*LinkLabel SecondLinkLabel*/
            SecondLinkLabel.Tag = "Orchestra Text Link"; /*Αλλαγή
*του Tag στο LinkLabel SecondLinkLabel*/
        }
        else if (FirstLinkLabel.Text == "Skene")
        {
            AncientGreekTheaterText.Visible = true; /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
            AncientGreekTheaterText.Text = "A large rectangular
building situated behind the orchestra, used as a backstage. Actors could
change their costumes and masks. Earlier the skene was a tent or hut,
later it became a permanent stone structure. These structures were
sometimes painted to serve as backdrops."; /*Αλλαγή κειμένου στο TextBox
*AncientGreekTheaterText*/

```

```

        AncientGreekTheaterText.Tag = "Ancient Greek Theater Text
- Skene"; //Αλλαγή του Tag του TextBox AncientGreekTheaterText
        TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); /*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
*Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
*μέγεθος γραμματοσειράς, στυλ γραμματοσειράς*/
        TextTitleLabel.Text = "Skene"; /*Αλλαγή κειμένου στο
*Label TextTitleLabel*/
        TextTitleLabel.Tag = "Text Title Label - Skene";
//Αλλαγή του Tag στο Label TextTitleLabel
        FirstLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*FirstLinkLabel*/
        FirstLinkLabel.Text = "Audience"; /*Αλλαγή Text στο
*LinkLabel FirstLinkLabel*/
        FirstLinkLabel.Tag = "Audience Text Link"; /*Αλλαγή του
*Tag στο LinkLabel FirstLinkLabel*/
        SecondLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*SecondLinkLabel*/
        SecondLinkLabel.Text = "Orchestra"; /*Αλλαγή Text στο
*LinkLabel SecondLinkLabel*/
        SecondLinkLabel.Tag = "Orchestra Text Link"; /*Αλλαγή
*του Tag στο LinkLabel SecondLinkLabel*/
    }
    else if (FirstLinkLabel.Text == "Orchestra")
    {
        AncientGreekTheaterText.Visible = true; /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
        AncientGreekTheaterText.Text = "Rising from the circle of
the orchestra was the audience. The theatres were originally built on a
very large scale to accommodate the large number of people on stage, as
well as the large number of people in the audience, up to fourteen
thousand."; //Αλλαγή κειμένου στο TextBox AncientGreekTheaterText
        AncientGreekTheaterText.Tag = "Ancient Greek Theater Text
- Orchestra"; //Αλλαγή του Tag του TextBox AncientGreekTheaterText
        TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); /*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
*Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
*μέγεθος γραμματοσειράς, στυλ γραμματοσειράς*/
        TextTitleLabel.Text = "Orchestra"; /*Αλλαγή κειμένου στο
*Label TextTitleLabel*/
        TextTitleLabel.Tag = "Text Title Label - Orchestra";
//Αλλαγή του Tag στο Label TextTitleLabel
        FirstLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*FirstLinkLabel*/
        FirstLinkLabel.Text = "Audience"; /*Αλλαγή Text στο
*LinkLabel FirstLinkLabel*/
        FirstLinkLabel.Tag = "Audience Text Link"; /*Αλλαγή του
*Tag στο LinkLabel FirstLinkLabel*/
        SecondLinkLabel.Visible = true; /*Εμφάνιση του LinkLabel
*SecondLinkLabel*/
        SecondLinkLabel.Text = "Skene"; /*Αλλαγή Text στο
*LinkLabel SecondLinkLabel*/
        SecondLinkLabel.Tag = "Skene Text Link"; /*Αλλαγή του
*Tag στο LinkLabel SecondLinkLabel*/
    }
}

private void SecondLinkLabel_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του LinkLabel SecondLinkLabel
{
    ReadTheTextButton.Visible = false; /*Απόκρυψη του Button
*ReadTheTextButton*/

```

```

        HideTheTextButton.Visible = true;    /*Εμφάνιση του Button
*HideTheTextButton*/
        increaseTextButton.Visible = true;    /*Εμφάνιση του button
*increaseTextButton*/
        decreaseTextButton.Visible = true;    /*Εμφάνιση του button
*decreaseTextButton*/

        /*
        * Χρήση εντολής if για τη λήψη απόφασης.
        * Ο έλεγχος αφορά το Text του LinkLabel SecondLinkLabel
        * Ανάλογα με το Text, θα εμφανίζονται τα παρακάτω αντικείμενα
με συγκεκριμένες ιδιότητες
        * που αφορούν το συγκεκριμένο θέμα (του link)
        */
        if (SecondLinkLabel.Text == "Audience")
        {
            AncientGreekTheaterText.Visible = true;    /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
            AncientGreekTheaterText.Text = "Rising from the circle of
the orchestra was the audience. The theatres were originally built on a
very large scale to accommodate the large number of people on stage, as
well as the large number of people in the audience, up to fourteen
thousand.";    //Αλλαγή κειμένου στο TextBox AncientGreekTheaterText
            AncientGreekTheaterText.Tag = "Ancient Greek Theater Text
- Audience";    //Αλλαγή του Tag του TextBox AncientGreekTheaterText
            TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold);    /*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
*Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
*μέγεθος γραμματοσειράς, στυλ γραμματοσειράς*/
            TextTitleLabel.Text = "Audience";    /*Αλλαγή κειμένου στο
*Label TextTitleLabel*/
            TextTitleLabel.Tag = "Text Title Label - Audience";
//Αλλαγή του Tag στο Label TextTitleLabel
            FirstLinkLabel.Visible = true;    /*Εμφάνιση του LinkLabel
*FirstLinkLabel*/
            FirstLinkLabel.Text = "Skene";    //Αλλαγή Text στο
*LinkLabel FirstLinkLabel*/
            FirstLinkLabel.Tag = "Skene Text Link";    /*Αλλαγή του Tag
*στο LinkLabel FirstLinkLabel*/
            SecondLinkLabel.Visible = true;    /*Εμφάνιση του LinkLabel
*SecondLinkLabel*/
            SecondLinkLabel.Text = "Orchestra";    /*Αλλαγή Text στο
*LinkLabel SecondLinkLabel*/
            SecondLinkLabel.Tag = "Orchestra Text Link";    /*Αλλαγή
*του Tag στο LinkLabel SecondLinkLabel*/
        }
        else if (SecondLinkLabel.Text == "Skene")
        {
            AncientGreekTheaterText.Visible = true;    /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
            AncientGreekTheaterText.Text = "A large rectangular
building situated behind the orchestra, used as a backstage. Actors could
change their costumes and masks. Earlier the skene was a tent or hut,
later it became a permanent stone structure. These structures were
sometimes painted to serve as backdrops.";    /*Αλλαγή κειμένου στο TextBox
*AncientGreekTheaterText*/
            AncientGreekTheaterText.Tag = "Ancient Greek Theater Text
- Skene";    //Αλλαγή του Tag του TextBox AncientGreekTheaterText
            TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold);    /*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
*Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
*μέγεθος γραμματοσειράς, στυλ γραμματοσειράς*/

```

```

        TextTitleLabel.Text = "Skene";    /*Αλλαγή κειμένου στο
*Label TextTitleLabel*/
        TextTitleLabel.Tag = "Text Title Label - Skene";
//Αλλαγή του Tag στο Label TextTitleLabel
        FirstLinkLabel.Visible = true;    /*Εμφάνιση του LinkLabel
*FirstLinkLabel*/
        FirstLinkLabel.Text = "Audience";    /*Αλλαγή Text στο
*LinkLabel FirstLinkLabel*/
        FirstLinkLabel.Tag = "Audience Text Link";    /*Αλλαγή του
*Tag στο LinkLabel FirstLinkLabel*/
        SecondLinkLabel.Visible = true;    /*Εμφάνιση του LinkLabel
*SecondLinkLabel*/
        SecondLinkLabel.Text = "Orchestra";    /*Αλλαγή Text στο
*LinkLabel SecondLinkLabel*/
        SecondLinkLabel.Tag = "Orchestra Text Link";    /*Αλλαγή
*του Tag στο LinkLabel SecondLinkLabel*/
    }
    else if (SecondLinkLabel.Text == "Orchestra")
    {
        AncientGreekTheaterText.Visible = true;    /*Εμφάνιση του
*TextBox AncientGreekTheaterText*/
        AncientGreekTheaterText.Text = "Rising from the circle of
the orchestra was the audience. The theatres were originally built on a
very large scale to accommodate the large number of people on stage, as
well as the large number of people in the audience, up to fourteen
thousand.";    //Αλλαγή κειμένου στο TextBox AncientGreekTheaterText
        AncientGreekTheaterText.Tag = "Ancient Greek Theater Text
- Orchestra";    //Αλλαγή του Tag του TextBox AncientGreekTheaterText
        TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold);    /*Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
*Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
*μέγεθος γραμματοσειράς, στυλ γραμματοσειράς*/
        TextTitleLabel.Text = "Orchestra";    /*Αλλαγή κειμένου στο
*Label TextTitleLabel*/
        TextTitleLabel.Tag = "Text Title Label - Orchestra";
//Αλλαγή του Tag στο Label TextTitleLabel
        FirstLinkLabel.Visible = true;    /*Εμφάνιση του LinkLabel
*FirstLinkLabel*/
        FirstLinkLabel.Text = "Audience";    /*Αλλαγή Text στο
*LinkLabel FirstLinkLabel*/
        FirstLinkLabel.Tag = "Audience Text Link";    /*Αλλαγή του
*Tag στο LinkLabel FirstLinkLabel*/
        SecondLinkLabel.Visible = true;    /*Εμφάνιση του LinkLabel
*SecondLinkLabel*/
        SecondLinkLabel.Text = "Skene";    /*Αλλαγή Text στο
*LinkLabel SecondLinkLabel*/
        SecondLinkLabel.Tag = "Skene Text Link";    /*Αλλαγή του
*Tag στο LinkLabel SecondLinkLabel*/
    }
}

private void HideTheTextButton_Click(object sender, EventArgs e)
{
    AncientGreekTheaterText.Visible = false;    /*Απόκρυψη του
*TextBox BasicInformationText*/
    ReadTheTextButton.Visible = true;    /*Εμφάνιση του Button
*ReadTheTextButton*/
    HideTheTextButton.Visible = false;    /*Απόκρυψη του Button
*HideTheTextButton*/
    FirstLinkLabel.Visible = false;    /*Απόκρυψη του LinkLabel
*FirstLinkLabel*/

```

```

        SecondLinkLabel.Visible = false; /*Απόκρυψη του LinkLabel
*SecondLinkLabel*/
        increaseTextButton.Visible = false; /*Απόκρυψη του button
*increaseTextButton*/
        decreaseTextButton.Visible = false; /*Απόκρυψη του button
*decreaseTextButton*/
    }

    private void increaseTextButton_Click_1(object sender, EventArgs
e)
    {
        float k = AncientGreekTheaterText.Font.Size; /*Δήλωση
*μεταβλητή για το Size της Font του TextBox AncientGreekTheaterText, τύπου
*float*/
        if (k <= 18) //Εντολή απόφασης για τον έλεγχο της τιμής του k
        {
            /* Κατασκευή αντικειμένου για το Font του TextBox
*BasicInformationText.
            * Στη 2η παράμετρο (fontSize) γίνεται αύξηση κατά 1em στο
*υπάρχον Size της Font του TextBox AncientGreekTheaterText
            */
            AncientGreekTheaterText.Font = new Font("Tahoma", k + 1,
FontStyle.Regular);
        }
    }

    private void decreaseTextButton_Click(object sender, EventArgs e)
    {
        float k = AncientGreekTheaterText.Font.Size; /*Δήλωση
*μεταβλητή για το Size της Font του TextBox AncientGreekTheaterText, τύπου
*float*/
        if (k >= 8) //Εντολή απόφασης για τον έλεγχο της τιμής του k
        {
            /* Κατασκευή αντικειμένου για το Font του TextBox
*BasicInformationText.
            * Στη 2η παράμετρο (fontSize) γίνεται μείωση κατά 1em στο
*υπάρχον Size της Font του TextBox AncientGreekTheaterText
            */
            AncientGreekTheaterText.Font = new Font("Tahoma", k - 1,
FontStyle.Regular);
        }
    }
}

```

## Κώδικας σχεδιασμού:

```

namespace Diploma_Work
{
    partial class AncientGreekTheater
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;
        private System.Windows.Forms.MainMenu mainMenu1;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
    }
}

```

```

    /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(AncientGreekTheater
));

    this.mainMenu1 = new System.Windows.Forms.MainMenu();
    this.OtherInfosMenu = new System.Windows.Forms.MenuItem();
    this.TheaterBuildingsTexts = new
System.Windows.Forms.MenuItem();
    this.TheaterBuildingsMenuItem = new
System.Windows.Forms.MenuItem();
    this.AudienceTextMenuItem = new
System.Windows.Forms.MenuItem();
    this.SkeneTextMenuItem = new System.Windows.Forms.MenuItem();
    this.OrchestraTextMenuItem = new
System.Windows.Forms.MenuItem();
    this.AncientGreekTheaterMenuItem = new
System.Windows.Forms.MenuItem();
    this.BackButton = new System.Windows.Forms.MenuItem();
    this.EpidaurusImage = new System.Windows.Forms.PictureBox();
    this.TextTitleLabel = new System.Windows.Forms.Label();
    this.AncientGreekTheaterText = new
System.Windows.Forms.TextBox();
    this.ReadTheTextButton = new System.Windows.Forms.Button();
    this.HideTheTextButton = new System.Windows.Forms.Button();
    this.SecondLinkLabel = new System.Windows.Forms.LinkLabel();
    this.FirstLinkLabel = new System.Windows.Forms.LinkLabel();
    this.increaseTextButton = new System.Windows.Forms.Button();
    this.decreaseTextButton = new System.Windows.Forms.Button();
    this.SuspendLayout();
    //
    // mainMenu1
    //
    this.mainMenu1.MenuItems.Add(this.OtherInfosMenu);
    this.mainMenu1.MenuItems.Add(this.BackButton);
    //
    // OtherInfosMenu
    //
    this.OtherInfosMenu.MenuItems.Add(this.TheaterBuildingsTexts);

this.OtherInfosMenu.MenuItems.Add(this.AncientGreekTheaterMenuItem);
    this.OtherInfosMenu.Text = "Other Infos";
    //
    // TheaterBuildingsTexts
    //

```

```

this.TheaterBuildingsTexts.MenuItems.Add(this.TheaterBuildingsMenuItem);

this.TheaterBuildingsTexts.MenuItems.Add(this.AudienceTextMenuItem);

this.TheaterBuildingsTexts.MenuItems.Add(this.SkeneTextMenuItem);

this.TheaterBuildingsTexts.MenuItems.Add(this.OrchestraTextMenuItem);
    this.TheaterBuildingsTexts.Text = "Theater Buildings";
    //
    // TheaterBuildingsMenuItem
    //
    this.TheaterBuildingsMenuItem.Text = "Theater Buildings";
    this.TheaterBuildingsMenuItem.Click += new
System.EventHandler(this.TheaterBuildingsMenuItem_Click);
    //
    // AudienceTextMenuItem
    //
    this.AudienceTextMenuItem.Text = "Audience";
    this.AudienceTextMenuItem.Click += new
System.EventHandler(this.AudienceTextMenuItem_Click);
    //
    // SkeneTextMenuItem
    //
    this.SkeneTextMenuItem.Text = "Skene";
    this.SkeneTextMenuItem.Click += new
System.EventHandler(this.SkeneTextMenuItem_Click);
    //
    // OrchestraTextMenuItem
    //
    this.OrchestraTextMenuItem.Text = "Orchestra";
    this.OrchestraTextMenuItem.Click += new
System.EventHandler(this.OrchestraTextMenuItem_Click);
    //
    // AncientGreekTheaterMenuItem
    //
    this.AncientGreekTheaterMenuItem.Text = "Ancient Greek
Theater";
    this.AncientGreekTheaterMenuItem.Click += new
System.EventHandler(this.AncientGreekTheaterMenuItem_Click);
    //
    // BackButton
    //
    this.BackButton.Text = "Back";
    this.BackButton.Click += new
System.EventHandler(this.BackButton_Click);
    //
    // EpidaurusImage
    //
    this.EpidaurusImage.Image =
((System.Drawing.Image)(resources.GetObject("EpidaurusImage.Image")));
    this.EpidaurusImage.Location = new System.Drawing.Point(0, 0);
    this.EpidaurusImage.Name = "EpidaurusImage";
    this.EpidaurusImage.Size = new System.Drawing.Size(240, 268);
    this.EpidaurusImage.Tag = "Epidaurus";
    //
    // TextTitleLabel
    //
    this.TextTitleLabel.Font = new System.Drawing.Font("Arial",
13F, System.Drawing.FontStyle.Bold);
    this.TextTitleLabel.Location = new System.Drawing.Point(0, 0);
    this.TextTitleLabel.Name = "TextTitleLabel";

```



```

        this.TextTitleLabel.Size = new System.Drawing.Size(240, 20);
        this.TextTitleLabel.Tag = "Text title label";
        this.TextTitleLabel.Text = "Ancient Greek Theater";
        this.TextTitleLabel.TextAlign =
System.Drawing.ContentAlignment.TopCenter;
        //
        // AncientGreekTheaterText
        //
        this.AncientGreekTheaterText.BackColor =
System.Drawing.Color.White;
        this.AncientGreekTheaterText.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
        this.AncientGreekTheaterText.Location = new
System.Drawing.Point(9, 20);
        this.AncientGreekTheaterText.Multiline = true;
        this.AncientGreekTheaterText.Name = "AncientGreekTheaterText";
        this.AncientGreekTheaterText.ReadOnly = true;
        this.AncientGreekTheaterText.ScrollBars =
System.Windows.Forms.ScrollBars.Vertical;
        this.AncientGreekTheaterText.Size = new
System.Drawing.Size(222, 174);
        this.AncientGreekTheaterText.TabIndex = 2;
        this.AncientGreekTheaterText.Tag = "Ancient greek theater
text";
        this.AncientGreekTheaterText.Text =
resources.GetString("AncientGreekTheaterText.Text");
        this.AncientGreekTheaterText.Visible = false;
        //
        // ReadTheTextButton
        //
        this.ReadTheTextButton.Location = new System.Drawing.Point(9,
244);
        this.ReadTheTextButton.Name = "ReadTheTextButton";
        this.ReadTheTextButton.Size = new System.Drawing.Size(222,
20);
        this.ReadTheTextButton.TabIndex = 3;
        this.ReadTheTextButton.Tag = "Read the text button";
        this.ReadTheTextButton.Text = "Read the Text";
        this.ReadTheTextButton.Click += new
System.EventHandler(this.ReadTheTextButton_Click);
        //
        // HideTheTextButton
        //
        this.HideTheTextButton.Location = new System.Drawing.Point(9,
244);
        this.HideTheTextButton.Name = "HideTheTextButton";
        this.HideTheTextButton.Size = new System.Drawing.Size(222,
20);
        this.HideTheTextButton.TabIndex = 6;
        this.HideTheTextButton.Tag = "Hide the text button";
        this.HideTheTextButton.Text = "Hide the Text";
        this.HideTheTextButton.Click += new
System.EventHandler(this.HideTheTextButton_Click);
        //
        // SecondLinkLabel
        //
        this.SecondLinkLabel.BackColor =
System.Drawing.Color.Transparent;
        this.SecondLinkLabel.Location = new System.Drawing.Point(9,
221);
        this.SecondLinkLabel.Name = "SecondLinkLabel";
        this.SecondLinkLabel.Size = new System.Drawing.Size(86, 20);

```

```

        this.SecondLinkLabel.TabIndex = 9;
        this.SecondLinkLabel.TabStop = false;
        this.SecondLinkLabel.Click += new
System.EventHandler(this.SecondLinkLabel_Click);
        //
        // FirstLinkLabel
        //
        this.FirstLinkLabel.BackColor =
System.Drawing.Color.Transparent;
        this.FirstLinkLabel.Location = new System.Drawing.Point(9,
197);
        this.FirstLinkLabel.Name = "FirstLinkLabel";
        this.FirstLinkLabel.Size = new System.Drawing.Size(86, 20);
        this.FirstLinkLabel.TabIndex = 10;
        this.FirstLinkLabel.TabStop = false;
        this.FirstLinkLabel.Click += new
System.EventHandler(this.FirstLinkLabel_Click);
        //
        // increaseTextButton
        //
        this.increaseTextButton.BackColor =
System.Drawing.SystemColors.GrayText;
        this.increaseTextButton.Font = new
System.Drawing.Font("Tahoma", 10F, System.Drawing.FontStyle.Bold);
        this.increaseTextButton.ForeColor =
System.Drawing.Color.Black;
        this.increaseTextButton.Location = new
System.Drawing.Point(169, 198);
        this.increaseTextButton.Name = "increaseTextButton";
        this.increaseTextButton.Size = new System.Drawing.Size(28,
20);
        this.increaseTextButton.TabIndex = 13;
        this.increaseTextButton.Tag = "Button for increasing the text
size";
        this.increaseTextButton.Text = "+A";
        this.increaseTextButton.Click += new
System.EventHandler(this.increaseTextButton_Click_1);
        //
        // decreaseTextButton
        //
        this.decreaseTextButton.BackColor =
System.Drawing.SystemColors.GrayText;
        this.decreaseTextButton.Font = new
System.Drawing.Font("Tahoma", 10F, System.Drawing.FontStyle.Bold);
        this.decreaseTextButton.ForeColor =
System.Drawing.Color.Black;
        this.decreaseTextButton.Location = new
System.Drawing.Point(203, 198);
        this.decreaseTextButton.Name = "decreaseTextButton";
        this.decreaseTextButton.Size = new System.Drawing.Size(28,
20);
        this.decreaseTextButton.TabIndex = 14;
        this.decreaseTextButton.Tag = "Button for decreasing the text
size";
        this.decreaseTextButton.Text = "-A";
        this.decreaseTextButton.Click += new
System.EventHandler(this.decreaseTextButton_Click);
        //
        // AncientGreekTheater
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(96F, 96F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Dpi;

```

```

        this.AutoScroll = true;
        this.ClientSize = new System.Drawing.Size(240, 268);
        this.Controls.Add(this.decreaseTextButton);
        this.Controls.Add(this.increaseTextButton);
        this.Controls.Add(this.FirstLinkLabel);
        this.Controls.Add(this.SecondLinkLabel);
        this.Controls.Add(this.HideTheTextButton);
        this.Controls.Add(this.ReadTheTextButton);
        this.Controls.Add(this.AncientGreekTheaterText);
        this.Controls.Add(this.TextTitleLabel);
        this.Controls.Add(this.EpidaurusImage);
        this.Menu = this.mainMenu1;
        this.Name = "AncientGreekTheater";
        this.Tag = "Ancient greek theater";
        this.Text = "Ancient Greek Theater";
        this.Activated += new
System.EventHandler(this.AncientGreekTheater_Activated);
        this.ResumeLayout(false);
    }

#endregion

private System.Windows.Forms.PictureBox EpidaurusImage;
private System.Windows.Forms.Label TextTitleLabel;
private System.Windows.Forms.TextBox AncientGreekTheaterText;
private System.Windows.Forms.Button ReadTheTextButton;
private System.Windows.Forms.MenuItem OtherInfosMenu;
private System.Windows.Forms.MenuItem TheaterBuildingsTexts;
private System.Windows.Forms.MenuItem AudienceTextMenuItem;
private System.Windows.Forms.MenuItem SkeneTextMenuItem;
private System.Windows.Forms.MenuItem OrchestraTextMenuItem;
private System.Windows.Forms.MenuItem AncientGreekTheaterMenuItem;
private System.Windows.Forms.Button BackButton;
private System.Windows.Forms.Button HideTheTextButton;
private System.Windows.Forms.LinkLabel SecondLinkLabel;
private System.Windows.Forms.LinkLabel FirstLinkLabel;
private System.Windows.Forms.MenuItem TheaterBuildingsMenuItem;
private System.Windows.Forms.Button increaseTextButton;
private System.Windows.Forms.Button decreaseTextButton;
    }
}

```

## **Form5 – Tragedy Writers:**

### **Κώδικας ενεργειών:**

```

using System;
using System.Linq;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Media;
using System.IO;

```

```

namespace Diploma_Work
{
    public partial class TragedyWriters : Form
    {
        MediaPlayer Media = new MediaPlayer();
        public TragedyWriters()
        {
            InitializeComponent();
        }

        void ObjectsVisibilityFalse()
        {
            TragedyWritersText.Visible = false; //Απόκρυψη του TextBox
            HideTheTextButton.Visible = false; //Απόκρυψη του Button
            FirstLink.Visible = false; //Απόκρυψη του LinkLabel
            SecondLink.Visible = false; //Απόκρυψη του LinkLabel
            increaseTextButton.Visible = false; //Απόκρυψη του button
            decreaseTextButton.Visible = false; //Απόκρυψη του button
        }

        private void TragedyWriters_Activated(object sender, EventArgs e)
        //Μέθοδος για το event Activated (ενεργοποίηση) της φόρμας TragedyWriters
        {
            ObjectsVisibilityFalse(); //Κλήση μεθόδου
            LifeText.Enabled = false; //Αφαίρεση της δυνατότητας
            επιλογής στο MenuItem LifeText
            WorksText.Enabled = false; //Αφαίρεση της δυνατότητας
            επιλογής στο MenuItem WorksText
            CommentaryText.Enabled = false; //Αφαίρεση της δυνατότητας
            επιλογής στο MenuItem CommentaryText
        }

        private void BackButton_Click(object sender, EventArgs e)
        //Μέθοδος για το event Click του MenuItem BackButton
        {
            ChoicesPage ChoicesPage = new ChoicesPage(); //Κατασκευή
            αντικειμένου ChoicesPage με εμβέλεια τοπική (μεθόδου)
            ChoicesPage.Show(); //Εμφάνιση της φόρμας ChoicesPage
            this.Hide(); //Αποκρυψη της παρούσας φόρμας
        }

        private void TragedyWriterChoiceCompoBox_SelectedIndexChanged
        (object sender, EventArgs e) //Μέθοδος για το event SelectedIndexChanged
        του CompoBox TragedyWriterChoiceCompoBox
        {
            LifeText.Enabled = true; //Απόκτηση της δυνατότητας επιλογής
            στο MenuItem LifeText
            WorksText.Enabled = true; //Απόκτηση της δυνατότητας
            επιλογής στο MenuItem WorksText
            CommentaryText.Enabled = true; //Απόκτηση της δυνατότητας
            επιλογής στο MenuItem CommentaryText
            /*
            * Αρχική διαγραφή των submenus WorkCategory1, WorkCategory2,
            WorkCategory3
            * που υπάρχουν by default (με parent το MenuItem WorksText)
            */
        }
    }
}

```

```

        * ώστε κάθε φορά που συμβαίνει το event Click στο CompoBox
TragedyWriterChoiceCompoBox
        * να υπάρχει η δυνατότητα να κατασκευάζονται νέα MenuItem
(με parent το WorksText)
        * ανάλογα με την επιλογή που πραγματοποιείται στο CompoBox
TragedyWriterChoiceCompoBox
        * (βλ. παρακάτω εντολές λήψης απόφασης if, else if, else)
        */
WorksText.MenuItems.Remove(WorkCategory1);
WorksText.MenuItems.Remove(WorkCategory2);
WorksText.MenuItems.Remove(WorkCategory3);
/*
        * Δημιουργία ελέγχου απόφασης if
        * Ανάλογα με την τιμή τύπου string που παίρνει το Property
Text του CompoBox TragedyWriterChoiceCompoBox
        * Συντελούνται οι παρακάτω ενέργειες
        */
        if (TragedyWriterChoiceCompoBox.Text == "Aeschylus")
//Περίπτωση όπου το Index του TragedyWriterChoiceCompoBox παίρνει την τιμή
"Aeshylus"
        {
            TragedyWritersImage.Image = Properties.Resources.Aeshylus;
//Αλλαγή της Image στο PictureBox TragedyWritersImage
            TragedyWritersImage.Tag = "Aeschylus"; //Αλλαγή του Tag
στο PictureBox TragedyWritersImage
            TextTitleLabel.Text = "Aeschylus"; //Αλλαγή του Text στο
Label TextTitleLabel
            TextTitleLabel.Font = new Font("Arial", 13,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
            TextTitleLabel.Tag = "Text Title Label - Aeschylus";
//Αλλαγή του Tag στο Label TextTitleLabel
            ReadTheTextButton.Visible = true; //Εμφάνιση του Button
ReadTheTextButton
        }
        else if (TragedyWriterChoiceCompoBox.Text == "Sophocles")
//Περίπτωση όπου το Index του TragedyWriterChoiceCompoBox παίρνει την τιμή
"Sophocles"
        {
            TragedyWritersImage.Image =
Properties.Resources.Sophocles; //Αλλαγή της Image στο PictureBox
TragedyWritersImage
            TragedyWritersImage.Tag = "Sophocles"; //Αλλαγή του Tag
στο PictureBox TragedyWritersImage
            TextTitleLabel.Text = "Sophocles"; //Αλλαγή του Text στο
Label TextTitleLabel
            TextTitleLabel.Font = new Font("Arial", 13,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
            TextTitleLabel.Tag = "Text Title Label - Sophocles";
//Αλλαγή του Tag στο Label TextTitleLabel
            ReadTheTextButton.Visible = true; //Εμφάνιση του Button
ReadTheTextButton
            WorksText.MenuItems.Add(WorkCategory1); //Δημιουργία
MenuItem με name WorkCategory1 με parent το MenuItem WorksText
            WorksText.MenuItems.Add(WorkCategory2); //Δημιουργία
MenuItem με name WorkCategory2 με parent το MenuItem WorksText
            WorkCategory1.Text = "The Theban Plays"; //Αλλαγή του
Text του MenuItem WorkCategory1

```

```

        WorkCategory2.Text = "Fragmentary Plays"; //Αλλαγή του
Text του MenuItem WorkCategory2
    }
    else //Περίπτωση όπου το Index του
TragedyWriterChoiceCompoBox παίρνει οποιαδήποτε άλλη τιμή
    {
        TragedyWritersImage.Image =
Properties.Resources.Euripides; //Αλλαγή της Image στο PictureBox
TragedyWritersImage
        TragedyWritersImage.Tag = "Euripides"; //Αλλαγή του Tag
στο PictureBox TragedyWritersImage
        TextTitleLabel.Text = "Euripides"; //Αλλαγή του Text στο
Label TextTitleLabel
        TextTitleLabel.Font = new Font("Arial", 13,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
        TextTitleLabel.Tag = "Text Title Label - Euripides";
//Αλλαγή του Tag στο Label TextTitleLabel
        ReadTheTextButton.Visible = true; //Εμφάνιση του Button
ReadTheTextButton
        WorksText.MenuItems.Add(WorkCategory1); //Δημιουργία
MenuItem με name WorkCategory1 με parent το MenuItem WorksText
        WorksText.MenuItems.Add(WorkCategory2); //Δημιουργία
MenuItem με name WorkCategory2 με parent το MenuItem WorksText
        WorksText.MenuItems.Add(WorkCategory3); //Δημιουργία
MenuItem με name WorkCategory3 με parent το MenuItem WorksText
        WorkCategory1.Text = "Tragedies"; //Αλλαγή του Text του
MenuItem WorkCategory1
        WorkCategory2.Text = "Fragmentary Tragedies"; //Αλλαγή
του Text του MenuItem WorkCategory2
        WorkCategory3.Text = "Satyr Play"; //Αλλαγή του Text του
MenuItem WorkCategory3
    }
}

private void ReadTheTextButton_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του Button ReadTheTextButton
{
    TragedyWritersText.Visible = true; //Εμφάνιση του TextBox
TragedyWritersText
    increaseTextButton.Visible = true; //Εμφάνιση του button
increaseTextButton
    decreaseTextButton.Visible = true; //Εμφάνιση του button
decreaseTextButton

    /*
    *Χρήση εντολής λήψης απόφασης if
    *ώστε ανάλογα με την τιμή τύπου string που παίρνει το Label
TextTitleLabel
    *το Property Text του TextBox TragedyWritersText να παίρνει
τη σχετική τιμή (κείμενο) τύπου string
    *όταν γίνεται το event Click του Button ReadTheTextButton
συντελείται
    */
    if (TextTitleLabel.Text == "Aeschylus") //Περίπτωση όπου το
Text του TextTitleLabel είναι "Aeshylus"
    {
        TragedyWriterChoiceCompoBox.Visible = false; //Απόκρυψη
του CompoBox TragedyWriterChoiceCompoBox
        TragedyWritersText.Text = "Aeschylus was an ancient Greek
playwright. His name derives from the Greek word αισχος (aischos), meaning

```

```

\"shame\". He is often recognized as the father of tragedy, and is the
earliest of the three Greek tragedians whose plays survive. According to
Aristotle, he expanded the number of characters in plays to allow for
conflict among them; previously, characters interacted only with the
chorus. Only seven of an estimated seventy to ninety plays by Aeschylus
have survived into modern times; there is an ongoing debate about the
authorship of one of these plays, Prometheus Bound. At least one of
Aeschylus' works was influenced by the Persian invasion of Greece, which
took place during his lifetime. His play The Persians remains a good
primary source of information about this period in Greek history. The war
was so important to the Greeks and to Aeschylus himself that, upon his
death around 456 BC, his epitaph commemorated his participation in the
Greek victory at Marathon rather than to his success as a playwright.";
//Αλλαγή του Text του TextBox TragedyWritersText
    TragedyWritersText.Tag = "Tragedy Writers Text- Aeschylus
General Information"; //Αλλαγή του Tag στο TextBox TragedyWritersText
}
else if (TextTitleLabel.Text == "Sophocles") //Περίπτωση
όπου το Text του TextTitleLabel είναι "Sophocles"
{
    TragedyWritersText.Text = "Sophocles was the second of the
three ancient Greek tragedians whose work has survived. His first plays
were written later than those of Aeschylus and earlier than those of
Euripides. According to the Suda, a 10th century encyclopedia, Sophocles
wrote 123 plays during the course of his life, but only seven have
survived in a complete form: Ajax, Antigone, Trachinian Women, Oedipus the
King, Electra, Philoctetes and Oedipus at Colonus. For almost 50 years,
Sophocles was the most-awarded playwright in the dramatic competitions of
the city-state of Athens that took place during the religious festivals of
the Lenaea and the Dionysia. Sophocles competed in around 30 competitions;
he won perhaps 24 and never received lower than second place; in
comparison, Aeschylus won 14 competitions and was defeated by Sophocles at
times, while Euripides won only 4 competitions. The most famous of
Sophocles' tragedies are those concerning Oedipus and Antigone: these are
often known as the Theban plays, although each play was actually a part of
different tetralogy, the other members of which are now lost. Sophocles
influenced the development of the drama, most importantly by adding a
third actor and thereby reducing the importance of the chorus in the
presentation of the plot. He also developed his characters to a greater
extent than earlier playwrights such as Aeschylus."; //Αλλαγή του Text
του TextBox TragedyWritersText
    TragedyWritersText.Tag = "Tragedy Writers Text- Sophocles
General Information"; //Αλλαγή του Tag στο TextBox TragedyWritersText
    TragedyWriterChoiceCompoBox.Visible = false; //Απόκρυψη
του CompoBox TragedyWriterChoiceCompoBox
}
else if (TextTitleLabel.Text == "Euripides") //Περίπτωση όπου
το Label TextTitleLabel έχει οποιαδήποτε άλλη τιμή
{
    TragedyWritersText.Text = "Euripides was the last of the
three great tragedians of classical Athens (the other two being Aeschylus
and Sophocles). Ancient scholars thought that Euripides had written
ninety-five plays, although four of those were probably written by
Critias. Eighteen or nineteen of Euripides' plays have survived complete.
There has been debate about his authorship of Rhesus, largely on stylistic
grounds and ignoring classical evidence that the play was his. Fragments,
some substantial, of most of the other plays also survive. More of his
plays have survived than those of Aeschylus and Sophocles together,
because of the unique nature of the Euripidean manuscript tradition.
Euripides is known primarily for having reshaped the formal structure of
Athenian tragedy by portraying strong female characters and intelligent
slaves and by satirizing many heroes of Greek mythology. His plays seem

```



```

modern by comparison with those of his contemporaries, focusing on the
inner lives and motives of his characters in a way previously unknown to
Greek audiences."; //Αλλαγή του Text του TextBox TragedyWritersText
    TragedyWritersText.Tag = "Tragedy Writers Text- Euripides
General Information"; //Αλλαγή του Tag στο TextBox TragedyWritersText
    TragedyWriterChoiceCompoBox.Visible = false; //Απόκρυψη
του CompoBox TragedyWriterChoiceCompoBox
    }

    ReadTheTextButton.Visible = false; //Απόκρυψη του Button
ReadTheTextButton
    HideTheTextButton.Visible = true; //Εμφάνιση του Button
HideTheTextButton
    }

    private void HideTheTextButton_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του Button HideTheTextButton
    {
        ObjectsVisibilityFalse(); //Κλήση μεθόδου
ObjectsVisibilityFalse
        TragedyWriterChoiceCompoBox.Visible = true; //Εμφάνιση του
CompoBox TragedyWriterChoiceCompoBox
        TragedyWritersImage.Image =
Properties.Resources.ThePlagueOfThebes; //Αλλαγή της εικόνας στο
PictureBox TragedyWritersImage
        TragedyWritersImage.Tag = "The plague of Thebes"; //Αλλαγή
του Tag στο PictureBox TragedyWritersImage
        TextTitleLabel.Text = "Choose a Tragedy Writer from the list
below"; //Αλλαγή του Text στο Label TextTitleLabel
        TextTitleLabel.Tag = "Text Title Label - Choose a Tragedy
Writer from the list below"; //Αλλαγή του Tag στο Label TextTitleLabel
        TextTitleLabel.Font = new Font("Arial", 8, FontStyle.Regular);
//Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
γραμματοσειράς, στυλ γραμματοσειράς
        LifeText.Enabled = false; //Αφαίρεση της δυνατότητας
επιλογής στο MenuItem LifeText
        WorksText.Enabled = false; //Αφαίρεση της δυνατότητας
επιλογής στο MenuItem WorksText
        CommentaryText.Enabled = false; //Αφαίρεση της δυνατότητας
επιλογής στο MenuItem CommentaryText
    }

    private void LifeText_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem LifeText
    {
        TragedyWritersText.Visible = true; //Εμφάνιση του TextBox
TragedyWritersText
        ReadTheTextButton.Visible = false; //Απόκρυψη του Button
ReadTheTextButton
        HideTheTextButton.Visible = true; //Εμφάνιση του Button
HideTheTextButton
        increaseTextButton.Visible = true; //Εμφάνιση του button
increaseTextButton
        decreaseTextButton.Visible = true; //Εμφάνιση του button
decreaseTextButton

        /*
        * Χρήση εντολής λήψης απόφασης if για τον έλεγχο του property
Text του CompoBox TragedyWriterChoiceCompoBox
        * Ανάλογα με την τιμή τύπου string που θα πάρει αυτό
        * θα πραγματοποιούνται οι παρακάτω εντολές

```

```

        * όταν το event Click του MenuItem LifeText συντελείται
        */
    if (TragedyWriterChoiceCompoBox.Text == "Aeschylus")
//Περίπτωση όπου το ComboBox TragedyWriterChoiceCompoBox παίρνει την τιμή
"Aeschylus"
    {
        TragedyWritersImage.Image = Properties.Resources.Aeshylus;
//Αλλαγή της Image στο PictureBox TragedyWritersImage
        TextTitleLabel.Text = "Aeschylus's Life"; //Αλλαγή του
Text του Label TextTitleLabel
        TextTitleLabel.Tag = "Title- Aeschylus's Life"; //Αλλαγή
του Tag του Label TextTitleLabel
        TextTitleLabel.Font = new Font("Arial", 11,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
        TragedyWritersText.Text = "There are no reliable sources
for the life of Aeschylus. He was said to have been born in c. 525 BC in
Eleusis, a small town about 27 kilometers northwest of Athens, which is
nestled in the fertile valleys of western Attica, though the date is most
likely based on counting back forty years from his first victory in the
Great Dionysia. His family was both wealthy and well established; his
father Euphorion was a member of the Eupatridae, the ancient nobility of
Attica. As a youth, he worked at a vineyard until, according to the 2nd-
century AD geographer Pausanias, the god Dionysus visited him in his sleep
and commanded him to turn his attention to the nascent art of tragedy. As
soon as he woke from the dream, the young Aeschylus began writing a
tragedy, and his first performance took place in 499 BC, when he was only
26 years old; He would eventually win his first victory at the City
Dionysia in 484 BC. In 510 BC, Cleomenes I (Aeschylus was 15 at the time)
expelled the sons of Peisistratus from Athens, which would lead
Cleisthenes to power. His reforms included a system of registration that
emphasized the importance of the deme over family tradition. In the last
decade of the 6th century, Aeschylus and his family were living in the
deme of Eleusina. The Persian Wars would play a large role in the
playwright's life and career. In 490 BC, Aeschylus and his brother
Cynegirus fought to defend Athens against Darius's invading Persian army
at the Battle of Marathon. The Athenians, though outnumbered, encircled
and slaughtered the Persian army. This pivotal defeat ended the first
Persian invasion of Greece proper and was celebrated across the city-
states of Greece. Though Athens was victorious, Cynegirus died in the
battle. He became a national hero when he was killed trying to draw a
retreating Persian ship back to shore. In 480, Aeschylus was called into
military service again, this time against Xerxes' invading forces at the
Battle of Salamis, and perhaps, too, at the Battle of Plataea in 479. Ion
of Chios was a witness for Aeschylus's war record and his contribution in
Salamis. Salamis holds a prominent place in The Persians, his oldest
surviving play, which was performed in 472 BC and won first prize at the
Dionysia. Aeschylus was one of many Greeks who had been initiated into the
Eleusinian Mysteries, a cult to Demeter based in his hometown of Eleusis.
As the name implies, members of the cult were supposed to have gained some
sort of mystical, secret knowledge. Firm details of the Mysteries'
specific rites are sparse, as members were sworn under the penalty of
death not to reveal anything about the Mysteries to non-initiates.
Nevertheless, according to Aristotle some thought that Aeschylus had
revealed some of the cult's secrets on stage. According to other sources,
an angry mob tried to kill Aeschylus on the spot, but he fled the scene.
Heracleides of Pontus recollected that the crowd watching the play tried
to stone Aeschylus. He then took refuge at the altar in the orchestra of
the Theater of Dionysus. When he stood trial for his offense he pleaded
ignorance. He was acquitted, with the jury sympathetic to the wounds that
Aeschylus and his brother Cynegirus suffered at Marathon. According to

```

the second-century A.D. author Aelian, Ameinias, Aeschylus's younger brother, helped his brother get acquitted by showing the jury the stump of the hand that Ameinias lost at Salamis, where he was voted bravest warrior. The truth is Cynegeirus lost his hand, and would later succumb to them at Marathon, and the award for bravery at Salamis went to Ameinias of Pallene, not Aeschylus's brother from Eleusis. Aeschylus traveled to Sicily once or twice in the 470s BC, having been invited by Hieron, tyrant of Syracuse, a major Greek city on the eastern side of the island; during one of these trips he produced *The Women of Aetna* (in honor of the city founded by Hieron) and restaged his *Persians*. By 473 BC, after the death of Phrynichus, one of his chief rivals, Aeschylus was the yearly favorite in the Dionysia, winning first prize in nearly every competition. In 472 BC, Aeschylus has staged four plays, all of them financed by an up and coming politician, Pericles. In 458 BC, he returned to Sicily for the last time, visiting the city of Gela where he died in 456 or 455 BC. After the success of *Persians*, he was invited personally by Hiero I of Syracuse to put the play on there. He also commissioned Aeschylus to write *Aetnaean Women* to celebrate the refounding of the city of Etna. *Aetnaean Women* celebrates Hieron, the savior of Hellenism in the West, for his founding of a new city. It is claimed that he was killed by a tortoise which fell out of the sky when dropped by an eagle, however this story is very likely apocryphal. Aeschylus' work was so respected by the Athenians that after his death, his were the only tragedies allowed to be restaged in subsequent competitions. His sons Euphorion and Euæon and his nephew Philocles would follow in his footsteps and become playwrights themselves. The inscription on Aeschylus' gravestone makes no mention of his theatrical renown, commemorating only his military achievements. Aeschylus married and had two sons, Euphorion and Euaeon (both of whom would become tragic poets). His nephew, Philocles (his sister's son), would also become a tragic poet. He had at least two brothers, Cynegeirus and Ameinias. "

```
//Αλλαγή του Text στο TextBox TragedyWritersText
TragedyWritersText.Tag = "Tragedy Writers Text - Life of
Aeschylus"; //Αλλαγή του Tag του TextBox TragedyWritersText
}
else if (TragedyWriterChoiceCompoBox.Text == "Sophocles")
//Περίπτωση όπου το ComboBox TragedyWriterChoiceCompoBox παίρνει την τιμή
"Sophocles"
{
    TragedyWritersImage.Image =
Properties.Resources.Sophocles; //Αλλαγή της Image στο PictureBox
TragedyWritersImage
    TextTitleLabel.Text = "Sophocles's Life"; //Αλλαγή του
Text του Label TextTitleLabel
    TextTitleLabel.Tag = "Title- Sophocles's Life"; //Αλλαγή
του Tag του Label TextTitleLabel
    TextTitleLabel.Font = new Font("Arial", 11,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
    TragedyWritersText.Text = "Sophocles, the son of
Sophillus, was a wealthy member of the rural deme (small community) of
Colonus Hippius in Attica, which would later become a setting for his
plays, and he was probably born there. His birth took place a few years
before the Battle of Marathon in 490 BC: the exact year is unclear,
although 497/6 is perhaps most likely. Sophocles' first artistic triumph
was in 468 BC when he took first prize in the Dionysia theatre competition
over the reigning master of Athenian drama, Aeschylus. According to
Plutarch the victory came under unusual circumstances. Instead of
following the custom of choosing judges by lot, the archon asked Cimon and
the other strategoi present to decide the victor of the contest. Plutarch
further contends that Aeschylus soon left for Sicily following this loss
to Sophocles. Although Plutarch says that this was Sophocles' first
```

production, it is now thought that this is an embellishment of the truth and that his first production was most likely in 470 BC. Triptolemus was probably one of the plays that Sophocles presented at this festival. Sophocles became a man of importance in the public halls of Athens as well as in the theatres. Sophocles was chosen to lead the paeon, a choral chant to a god, at the age of 16 celebrating the decisive Greek sea victory over the Persians at the Battle of Salamis. This rather insufficient information about Sophocles' civic life implies he was a well-liked man who participated in activities in society and showed remarkable artistic ability. He was also elected as one of ten strategoi, high executive officials that commanded the armed forces, as a junior colleague of Pericles. Sophocles was born extremely wealthy (his father was a wealthy armour manufacturer) and was highly educated throughout his entire life. Early in his career, the politician Cimon might have been one of his patrons, although if he was there was no ill will borne by Pericles, Cimon's rival, when Cimon was ostracized in 461 BC. In 443/2 he served as one of the Hellenotamiai, or treasurers of Athena, helping to manage the finances of the city during the political ascendancy of Pericles. According to the Vita Sophoclis he served as a general in the Athenian campaign against Samos, which had revolted in 441 BC; he was supposed to have been elected to his post as the result of his production of Antigone. In 420 he welcomed and set up an altar for the icon of Asclepius at his house, when the deity was introduced in Athens. For this he was given the posthumous epithet Dexion (receiver) by the Athenians. He was also elected, in 413 BC, to be one of the commissioners crafting a response to the catastrophic destruction of the Athenian expeditionary force in Sicily during the Peloponnesian War. Sophocles died at the age of ninety or ninety-one in the winter of 406/5 BC, having seen within his lifetime both the Greek triumph in the Persian Wars and the terrible bloodletting of the Peloponnesian War. As with many famous men in classical antiquity, Sophocles' death inspired a number of apocryphal stories about the cause. Perhaps the most famous is the suggestion that he died from the strain of trying to recite a long sentence from his Antigone without pausing to take a breath; another account suggests he choked while eating grapes at the Anthesteria festival in Athens. A third account holds that he died of happiness after winning his final victory at the City Dionysia. A few months later, the comic poet wrote this eulogy in his play titled The Muses: "Blessed is Sophocles, who had a long life, was a man both happy and talented, and the writer of many good tragedies; and he ended his life well without suffering any misfortune." This is somewhat ironic, for according to some accounts his own sons tried to have him declared incompetent near the end of his life; he is said to have refuted their charge in court by reading from his as yet unproduced Oedipus at Colonus. Both Iophon, one of his sons, and a grandson, also called Sophocles, followed in his footsteps to become playwrights. "; //Αλλαγή του Text

```

στο TextBox TragedyWritersText
        TragedyWritersText.Tag = "Tragedy Writers Text - Life of
Sophocles"; //Αλλαγή του Tag του TextBox TragedyWritersText
    }
    else //Περίπτωση όπου το ComboBox
TragedyWriterChoiceComboBox οποιαδήποτε άλλη τιμή
    {
        TragedyWritersImage.Image =
Properties.Resources.Euripides; //Αλλαγή της Image στο PictureBox
TragedyWritersImage
        TragedyWritersText.Visible = true; //Εμφάνιση του
TextBox TragedyWritersText
        TextTitleLabel.Text = "Euripides's Life"; //Αλλαγή του
Text του Label TextTitleLabel
        TextTitleLabel.Tag = "Title- Euripides's Life"; //Αλλαγή
του Tag του Label TextTitleLabel

```

```

        TextTitleLabel.Font = new Font("Arial", 11,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
        TragedyWritersText.Text = "Little is known about
Euripides, and most recorded sources are based on legend and hearsay.
According to one legend, Euripides was born in Salamís on 23 September 480
BC, the day of the Persian War's greatest naval battle. Other sources
estimate that he was born as early as 485 BC. His father's name was either
Mnesarchus or Mnesarchides and his mother's name was Cleito. Evidence
suggests that the family was wealthy and influential. It is recorded that
he served as a cup-bearer for Apollo's dancers, but he grew to question
the religion he grew up with, exposed as he was to thinkers such as
Protagoras, Socrates, and Anaxagoras. He was married twice, to Choerile
and Melito, though sources disagree as to which woman he married first. He
had three sons and it is rumored that he also had a daughter who was
killed after a rabid dog attacked her (some say this was merely a joke
made by Aristophanes, who often poked fun at Euripides). The record of
Euripides' public life, other than his involvement in dramatic
competitions, is almost non-existent. The only reliable story of note is
one by Aristotle about Euripides' involvement in a dispute over a liturgy
(an account that offers strong evidence that Euripides was a wealthy man).
It has been said that he traveled to Syracuse, Sicily; that he engaged in
various public or political activities during his lifetime; that he wrote
his tragedies in a sanctuary, The Cave of Euripides on Salamis Island; and
that he left Athens at the invitation of King Archelaus I of Macedon and
stayed with him in Macedonia and allegedly died there in 406 B.C. after
being accidentally attacked by the king's hunting dogs while walking in
the woods. According to Pausanias, Euripides was buried in Macedonia.";
//Αλλαγή του Text στο TextBox TragedyWritersText
        TragedyWritersText.Tag = "Tragedy Writers Text - Life of
Euripides"; //Αλλαγή του Tag του TextBox TragedyWritersText
    }
}

private void WorksText_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem WorksText
{
    ReadTheTextButton.Visible = false; //Απόκρυψη του Button
ReadTheTextButton
    HideTheTextButton.Visible = true; //Εμφάνιση του Button
HideTheTextButton
    TextTitleLabel.Text = "Aeschylus's Works"; //Αλλαγή του Text
του Label TextTitleLabel
    TextTitleLabel.Tag = "Title- Aeschylus's Works"; //Αλλαγή
του Tag του Label TextTitleLabel
    TextTitleLabel.Font = new Font("Arial", 11, FontStyle.Bold);
//Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
γραμματοσειράς, στυλ γραμματοσειράς
    TragedyWritersText.Visible = true; //Εμφάνιση του TextBox
TragedyWritersText
    TragedyWritersText.Text = "The Greek art of the drama had its
roots in religious festivals for the gods, chiefly Dionysus, the god of
wine. During Aeschylus' lifetime, dramatic competitions became part of the
City Dionysia in the spring. The festival began with an opening
procession, continued with a competition of boys singing dithyrambs, and
culminated in a pair of dramatic competitions. The first competition,
which Aeschylus would have participated in, was for the tragedians, and
consisted of three playwrights each presenting three tragic plays followed
by a shorter comedic satyr play. A second competition of five comedic
playwrights followed, and the winners of both competitions were chosen by

```

a panel of judges. Aeschylus entered many of these competitions in his lifetime, and various ancient sources attribute between seventy and ninety plays to him. Only seven tragedies have survived intact: *The Persians*, *Seven against Thebes*, *The Suppliants*, the trilogy known as *The Oresteia*, consisting of the three tragedies *Agamemnon*, *The Libation Bearers* and *The Eumenides*, and *Prometheus Bound* (whose authorship is disputed). With the exception of this last play—the success of which is uncertain—all of Aeschylus' extant tragedies are known to have won first prize at the City Dionysia. The Alexandrian *Life of Aeschylus* indicates that the playwright took the first prize at the City Dionysia thirteen times. This compares favorably with Sophocles' reported eighteen victories (with a substantially larger catalogue, at an estimated 120 plays), and dwarfs the five victories of Euripides (who featured a catalogue of roughly 90 plays). One hallmark of Aeschylean dramaturgy appears to have been his tendency to write connected trilogies in which each play serves as a chapter in a continuous dramatic narrative. The *Oresteia* is the only wholly extant example of this type of connected trilogy, but there is ample evidence that Aeschylus wrote such trilogies often. The comic satyr plays that would follow his dramatic trilogies often treated a related mythic topic. For example, the *Oresteia*'s satyr play *Proteus* treated the story of Menelaus's detour in Egypt on his way home from the Trojan War. Based on the evidence provided by a catalogue of Aeschylean play titles, scholia, and play fragments recorded by later authors, it is assumed that three other of Aeschylus' extant plays were components of connected trilogies: *Seven against Thebes* being the final play in an Oedipus trilogy, and *The Suppliants* and *Prometheus Bound* each being the first play in a Danaid trilogy and *Prometheus* trilogy, respectively (see below). Scholars have moreover suggested several completely lost trilogies derived from known play titles. A number of these trilogies treated myths surrounding the Trojan War. One—collectively called the *Achilleis* and comprising the titles *Myrmidons*, *Nereids* and *Phrygians* (alternately, *The Ransoming of Hector*). *Myrmidons* was based on books 9 and 16 in Homer's *Iliad*. The plot follows Achilles, who sits in silent indignation over his humiliation at Agamemnon's hands for most of the play. Envoys from the Greek army attempt to reconcile him to Agamemnon, but he yields only to his friend and lover Patroclus, who then battles the Trojans in Achilles' armor. The bravery and death of Patroclus are reported in a messenger's speech, which is followed by mourning. *Nereids* was based on books 18, 19, and 22 of the *Iliad*, follows the Daughters of Nereus, the sea god, lament Patroclus's death. In this play a messenger tells how Achilles, perhaps reconciled to Agamemnon and the Greeks, slew Hector. In *Phrygians*, Achilles sits in silent mourning over Patroclus, after a brief discussion with Hermes. Hermes then brings in King Priam of Troy, who wins over Achilles and ransoms his son's body in a spectacular coup de théâtre. An enormous scale is brought on stage and Hector's body is placed in one scale and gold in the other. The dynamic dancing of the chorus of Trojans when they enter with Priam is reported by Aristophanes. One of his lost plays, *Niobe* the heroine's children have been slain by Apollo and Artemis because Niobe had gloated that she had more children than their mother, Leto. Niobe sits in silent mourning on stage during most of the play. In the *Republic*, Plato called the line "God plants a fault in mortals when he wills to destroy a house utterly." immoral. Another trilogy apparently recounts the entry of the Trojan ally Memnon into the war, and his death at the hands of Achilles (*Memnon* and *The Weighing of Souls* being two components of the trilogy); *The Award of the Arms*, *The Phrygian Women*, and *The Salaminian Women* suggest a trilogy about the madness and subsequent suicide of the Greek hero Ajax; Aeschylus also seems to have treated Odysseus' return to Ithaca after the war (including his killing of his wife Penelope's suitors and its consequences) with a trilogy consisting of *The Soul-raisers*, *Penelope* and *The Bone-gatherers*. Other suggested trilogies touched on the myth of Jason and the Argonauts (*Argô*, *Lemnian Women*, *Hypsipylê*); the life of Perseus (*The Net-draggers*, *Polydektês*,



```

Phorkides); the birth and exploits of Dionysus (Semele, Bacchae,
Pentheus); and the aftermath of the war portrayed in Seven against Thebes
(Eleusinians, Argives (or Argive Women), Sons of the Seven). Only seven
tragedies have survived intact: The Persians, Seven against Thebes, The
Suppliants, the trilogy known as The Oresteia, consisting of the three
tragedies Agamemnon, The Libation Bearers and The Eumenides, and
Prometheus Bound (whose authorship is disputed)."; //Αλλαγή του Text στο
TextBox TragedyWritersText
    TragedyWritersText.Tag = "Tragedy Writers Text - Aeschylus's
Works"; //Αλλαγή του Tag του TextBox TragedyWritersText
    TragedyWritersImage.Image = Properties.Resources.Aeshylus;
//Αλλαγή της Image στο PictureBox TragedyWritersImage
    increaseTextButton.Visible = true; //Εμφάνιση του button
increaseTextButton
    decreaseTextButton.Visible = true; //Εμφάνιση του button
decreaseTextButton
}

private void WorkCategory1_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem WorkCategory1
{
    ReadTheTextButton.Visible = false; //Απόκρυψη του Button
ReadTheTextButton
    HideTheTextButton.Visible = true; //Εμφάνιση του Button
HideTheTextButton
    TragedyWritersText.Visible = true; //Εμφάνιση του TextBox
TragedyWritersText
    increaseTextButton.Visible = true; //Εμφάνιση του button
increaseTextButton
    decreaseTextButton.Visible = true; //Εμφάνιση του button
decreaseTextButton

    if (WorkCategory1.Text == "The Theban Plays") //Δημιουργία
εντολής ελέγχου απόφασης για τον έλεγχο του κειμένου του MenuItem
WorkCategory1
    {
        TextTitleLabel.Text = "Sophocles's Theban Plays";
//Αλλαγή του Text του Label TextTitleLabel
        TextTitleLabel.Tag = "Title- Sophocles's Work \"The Theban
Plays\""; //Αλλαγή του Tag του Label TextTitleLabel
        TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
        SecondLink.Visible = true; //Εμφάνιση του LinkLabel
SecondLink
        SecondLink.Text = "Fragmentary Plays"; //Αλλαγή του Text
του LinkLabel SecondLink
        SecondLink.Tag = "Link to Sophocles's\"Fragmentary Plays\"
Text"; //Αλλαγή του Tag του LinkLabel SecondLink
        TragedyWritersText.Text = "The Theban plays consist of
three plays: Antigone, Oedipus the King (also called Oedipus Tyrannus or
Oedipus Rex), and Oedipus at Colonus. All three plays concern the fate of
Thebes during and after the reign of King Oedipus. They have often been
published under a single cover. Sophocles, however, wrote the three plays
for separate festival competitions, many years apart. Not only are the
Theban plays not a true trilogy (three plays presented as a continuous
narrative) but they are not even an intentional series and contain some
inconsistencies among them. He also wrote other plays having to do with
Thebes, such as The Progeny, of which only fragments have survived. Other
than the three Theban plays, there are four surviving plays by Sophocles:

```



```

Ajax, The Trachiniae, Electra, and Philoctetes, the last of which won
first prize."; //Αλλαγή του Text στο TextBox TragedyWritersText
    TragedyWritersText.Tag = "Tragedy Writers Text-
Sophocles's \"The Theban Plays\" Works"; //Αλλαγή του Tag του TextBox
TragedyWritersText
    TragedyWritersImage.Image =
Properties.Resources.Sophocles; //Αλλαγή της Image στο PictureBox
TragedyWritersImage
    }
    else
    {
        TextTitleLabel.Text = "Euripides's Tragedies"; //Αλλαγή
του Text του Label TextTitleLabel
        TextTitleLabel.Tag = "Title- Euripides's Tragedies";
//Αλλαγή του Tag του Label TextTitleLabel
        TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
        FirstLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
        FirstLink.Text = "Fragmentary Tragedies"; //Αλλαγή του
Text του LinkLabel FirstLink
        FirstLink.Tag = "Link to Euripides\"Fragmentary
Tragedies\" Text"; //Αλλαγή του Tag του LinkLabel FirstLink
        SecondLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
        SecondLink.Text = "Satyr Play"; //Αλλαγή του Text του
LinkLabel FirstLink
        SecondLink.Tag = "Link to Euripides\"Satyr Play\" Text";
//Αλλαγή του Tag του LinkLabel FirstLink
        TragedyWritersText.Text = "1. Alcestis (438 BC, second
prize) \r\n2. Medea (431 BC, third prize) \r\n3. Heracleidae (c. 430 BC)
\r\n4. Hippolytus (428 BC, first prize) \r\n5. Andromache (c. 425 BC)
\r\n6. Hecuba (c. 424 BC) \r\n7. The Suppliants (c. 423 BC) \r\n8. Electra
(c. 420 BC) \r\n9. Heracles (c. 416 BC) \r\n10. The Trojan Women (415 BC,
second prize) \r\n11. Iphigenia in Tauris (c. 414 BC) \r\n12. Ion (c. 414
BC) \r\n13. Helen (412 BC) \r\n14. Phoenician Women (c. 410 BC) \r\n15.
Orestes (408 BC) \r\n16. Bacchae and Iphigenia at Aulis (405 BC,
posthumous, first prize) \r\n17. Rhesus (uncertain date)"; //Αλλαγή του
Text στο TextBox TragedyWritersText
        TragedyWritersText.Tag = "Tragedy Writers Text -
Euripides's \"Tragedies\""; //Αλλαγή του Tag του TextBox
TragedyWritersText
        TragedyWritersImage.Image =
Properties.Resources.Euripides; //Αλλαγή της Image στο PictureBox
TragedyWritersImage
    }
}

private void WorkCategory2_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem WorkCategory2
{
    ReadTheTextButton.Visible = false; //Απόκρυψη του Button
ReadTheTextButton
    HideTheTextButton.Visible = true; //Εμφάνιση του Button
HideTheTextButton
    TragedyWritersText.Visible = true; //Εμφάνιση του TextBox
TragedyWritersText
    increaseTextButton.Visible = true; //Εμφάνιση του button
increaseTextButton

```

```

        decreaseTextButton.Visible = true; //Εμφάνιση του button
decreaseTextButton

/*
 * Χρήση εντολής λήψης απόφασης if για τον έλεγχο του property
Text του MenuItem WorkCategory2
 * Ανάλογα με την τιμή που θα έχει αυτό
 * Θα πραγματοποιούνται οι παρακάτω εντολές
 */
if (WorkCategory2.Text == "Fragmentary Plays")
{
    TextTitleLabel.Text = "Sophocles's Fragmentary Plays";
//Αλλαγή του Text του Label TextTitleLabel
    TextTitleLabel.Tag = "Title- Sophocles's Work
\"Fragmentary Plays\""; //Αλλαγή του Tag του Label TextTitleLabel
    TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
    SecondLink.Visible = true; //Εμφάνιση του LinkLabel
SecondLink
    SecondLink.Text = "The Theban Plays"; //Αλλαγή του Text
του LinkLabel SecondLink
    SecondLink.Tag = "Link to Sophocles's \"The Theban Plays\"
Text"; //Αλλαγή του Tag του LinkLabel SecondLink
    TragedyWritersText.Text = "Fragments of The Tracking
Satyrs (Ichneutae) were discovered in Egypt in 1907. These amount to about
half of the play, making it the best preserved satyr play after Euripides'
Cyclops, which survives in its entirety. Fragments of The Progeny
(Epigonoi) were discovered in April 2005 by classicists at Oxford
University with the help of infrared technology previously used for
satellite imaging. The tragedy tells the story of the second siege of
Thebes. A number of other Sophoclean works have survived only in
fragments, including: \r\n• Aias Lokros (Ajax the Locrian) \r\n• Akhaiôn
Syllogos (The Gathering of the Achaeans) \r\n• Aleadae (The Sons of Aleus)
\r\n• Creusa \r\n• Eurypylos \r\n• Hermione \r\n• Inachos \r\n• Lacaenae
(Lacaenian Women) \r\n• Manteis or Polyidus (The Prophets or Polyidus)
\r\n• Nauplios Katapleon (Nauplius' Arrival) \r\n• Nauplios Pyrkaeus
(Nauplius' Fires) \r\n• Niobe \r\n• Oeneus \r\n• Oenomaus \r\n• Poimenes
(The Shepherds) \r\n• Polyxene \r\n• Syndeipnoi (The Diners, or, The
Banqueters) \r\n• Tereus \r\n• Thyestes r\n• Troilus \r\n• Phaedra \r\n•
Triptolemus \r\n• Tyro Keiromene (Tyro Shorn) \r\n• Tyro Anagnorizomene
(Tyro Rediscovered)."; //Αλλαγή του Text στο TextBox TragedyWritersText
    TragedyWritersText.Tag = "Tragedy Writers Text-
Sophocles's \"Fragmentary Plays\""; //Αλλαγή του Tag του TextBox
TragedyWritersText
    TragedyWritersImage.Image =
Properties.Resources.Sophocles; //Αλλαγή της Image στο PictureBox
TragedyWritersImage
}
else
{
    TextTitleLabel.Text = "Euripides's Fragmentary Tragedies";
//Αλλαγή του Text του Label TextTitleLabel
    TextTitleLabel.Tag = "Title- Euripides's Fragmentary
Tragedies"; //Αλλαγή του Tag του Label TextTitleLabel
    TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
    FirstLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink

```

```

        FirstLink.Text = "Tragedies"; //Αλλαγή του Text του
LinkLabel FirstLink
        FirstLink.Tag = "Link to Euripides \"Tragedies\" Text";
//Αλλαγή του Tag του LinkLabel FirstLink
        SecondLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
        SecondLink.Text = "Satyr Play"; //Αλλαγή του Text του
LinkLabel FirstLink
        SecondLink.Tag = "Link to Euripides \"Satyr Play\" Text";
//Αλλαγή του Tag του LinkLabel FirstLink
        TragedyWritersText.Text = "The following plays have come
down to us today only in fragmentary form; some consist of only a handful
of lines, but with some the fragments are extensive enough to allow
tentative reconstruction. \r\n1. Telephus (438 BC) \r\n2. Cretans (c. 435
BC) \r\n3. Stheneboea (before 429 BC) \r\n4. Bellerophon (c. 430 BC)
\r\n5. Cresphontes (ca. 425 BC) \r\n6. Erechtheus (422 BC) \r\n7. Phaethon
(c. 420 BC) \r\n8. Wise Melanippe (c. 420 BC) \r\n9. Alexandros (415
BC) \r\n10. Palamedes (415 BC) \r\n11. Sisyphus (415 BC) \r\n12. Captive
Melanippe (412 BC) \r\n13. Andromeda (412 BC with Euripides' Helen)
\r\n14. Antiope (c. 410 BC) \r\n15. Archelaus (c. 410 BC) \r\n16.
Hypsipyle (c. 410 BC) \r\n17. Philoctetes (c. 410 BC)"; //Αλλαγή του Text
στο TextBox TragedyWritersText
        TragedyWritersText.Tag = "Tragedy Writers Text-
Euripides's \"Fragmentary Tragedies\""; //Αλλαγή του Tag του TextBox
TragedyWritersText
        TragedyWritersImage.Image =
Properties.Resources.Euripides; //Αλλαγή της Image στο PictureBox
TragedyWritersImage
    }
}

private void WorkCategory3_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem WorkCategory3
{
    ReadTheTextButton.Visible = false; //Απόκρυψη του Button
ReadTheTextButton
    HideTheTextButton.Visible = true; //Εμφάνιση του Button
HideTheTextButton
    TragedyWritersText.Visible = true; //Εμφάνιση του TextBox
TragedyWritersText
    TextTitleLabel.Text = "Euripides's Satyr Play"; //Αλλαγή του
Text του Label TextTitleLabel
    TextTitleLabel.Tag = "Title- Euripides's \"Satyr Play\"";
//Αλλαγή του Tag του Label TextTitleLabel
    TextTitleLabel.Font = new Font("Arial", 10, FontStyle.Bold);
//Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
γραμματοσειράς, στυλ γραμματοσειράς
    FirstLink.Visible = true; //Εμφάνιση του LinkLabel FirstLink
    FirstLink.Text = "Tragedies"; //Αλλαγή του Text του
LinkLabel FirstLink
    FirstLink.Tag = "Link to Euripides \"Tragedies\" Text";
//Αλλαγή του Tag του LinkLabel FirstLink
    SecondLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
    SecondLink.Text = "Fragmentary Tragedies"; //Αλλαγή του Text
του LinkLabel FirstLink
    SecondLink.Tag = "Link to Euripides \"Fragmentary Tragedies\"
Text"; //Αλλαγή του Tag του LinkLabel FirstLink
    TragedyWritersText.Text = "1. Cyclops (uncertain date)";
    TragedyWritersText.Tag = "Tragedy Writers Text- Euripides
\"Satyr Play\""; //Αλλαγή του Tag του TextBox TragedyWritersText

```

```

        increaseTextButton.Visible = true; //Εμφάνιση του button
increaseTextButton
        decreaseTextButton.Visible = true; //Εμφάνιση του button
decreaseTextButton
    }

    private void FirstLink_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του LinkLabel FirstLink
    {
        /*
        * Δημιουργία εντολής λήψης απόφασης για τον έλεγχο του Text
του LinkLabel FirstLink.
        * Ανάλογα με ποιο θα είναι αυτό (το Text)
        * Θα εκτελούνται οι παρακάτω εντολές
        */
        if (FirstLink.Text == "Fragmentary Tragedies")
        {
            TextTitleLabel.Text = "Euripides's Fragmentary Tragedies";
//Αλλαγή του Text του Label TextTileLabel
            TextTitleLabel.Tag = "Title- Euripides's Fragmentary
Tragedies"; //Αλλαγή του Tag του Label TextTileLabel
            TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
            FirstLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
            FirstLink.Text = "Tragedies"; //Αλλαγή του Text του
LinkLabel FirstLink
            FirstLink.Tag = "Link to Euripides \"Tragedies\" Text";
//Αλλαγή του Tag του LinkLabel FirstLink
            SecondLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
            SecondLink.Text = "Satyr Play"; //Αλλαγή του Text του
LinkLabel FirstLink
            SecondLink.Tag = "Link to Euripides \"Satyr Play\" Text";
//Αλλαγή του Tag του LinkLabel FirstLink
            TragedyWritersText.Text = "The following plays have come
down to us today only in fragmentary form; some consist of only a handful
of lines, but with some the fragments are extensive enough to allow
tentative reconstruction. \r\n1. Telephus (438 BC) \r\n2. Cretans (c. 435
BC) \r\n3. Stheneboea (before 429 BC) \r\n4. Bellerophon (c. 430 BC)
\r\n5. Cresphontes (ca. 425 BC) \r\n6. Erechtheus (422 BC) \r\n7. Phaethon
(c. 420 BC) \r\n8. Wise Melanippe (c. 420 BC) \r\n9. Alexandros (415 BC)
\r\n10. Palamedes (415 BC) \r\n11. Sisyphus (415 BC) \r\n12. Captive
Melanippe (412 BC) \r\n13. Andromeda (412 BC with Euripides' Helen)
\r\n14. Antiope (c. 410 BC) \r\n15. Archelaus (c. 410 BC) \r\n16.
Hypsipyle (c. 410 BC) \r\n17. Philoctetes (c. 410 BC)"; //Αλλαγή του Text
στο TextBox TragedyWritersText
            TragedyWritersText.Tag = "Tragedy Writers Text-
Euripides's \"Fragmentary Tragedies\""; //Αλλαγή του Tag του TextBox
TragedyWritersText
        }
        else
        {
            TextTitleLabel.Text = "Euripides's Tragedies"; //Αλλαγή
του Text του Label TextTileLabel
            TextTitleLabel.Tag = "Title- Euripides's Tragedies";
//Αλλαγή του Tag του Label TextTileLabel
            TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του

```

```

Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
    FirstLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
    FirstLink.Text = "Fragmentary Tragedies"; //Αλλαγή του
Text του LinkLabel FirstLink
    FirstLink.Tag = "Link to Euripides\"Fragmentary
Tragedies\" Text"; //Αλλαγή του Tag του LinkLabel FirstLink
    SecondLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
    SecondLink.Text = "Satyr Play"; //Αλλαγή του Text του
LinkLabel FirstLink
    SecondLink.Tag = "Link to Euripides\"Satyr Play\" Text";
//Αλλαγή του Tag του LinkLabel FirstLink
    TragedyWritersText.Text = "1. Alcestis (438 BC, second
prize) \r\n2. Medea (431 BC, third prize) \r\n3. Heracleidae (c. 430 BC)
\r\n4. Hippolytus (428 BC, first prize) \r\n5. Andromache (c. 425 BC)
\r\n6. Hecuba (c. 424 BC) \r\n7. The Suppliants (c. 423 BC) \r\n8. Electra
(c. 420 BC) \r\n9. Heracles (c. 416 BC) \r\n10. The Trojan Women (415 BC,
second prize) \r\n11. Iphigenia in Tauris (c. 414 BC) \r\n12. Ion (c. 414
BC) \r\n13. Helen (412 BC) \r\n14. Phoenician Women (c. 410 BC) \r\n15.
Orestes (408 BC) \r\n16. Bacchae and Iphigenia at Aulis (405 BC,
posthumous, first prize) \r\n17. Rhesus (uncertain date)"; //Αλλαγή του
Text στο TextBox TragedyWritersText
    TragedyWritersText.Tag = "Tragedy Writers Text -
Euripides's \"Tragedies\""; //Αλλαγή του Tag του TextBox
TragedyWritersText
    }
}

private void SecondLink_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του LinkLabel SecondLink
{
    /*
    * Δημιουργία εντολής λήψης απόφασης για τον έλεγχο του Text
του LinkLabel SecondLink.
    * Ανάλογα με ποιο θα είναι αυτό (το Text)
    * θα εκτελούνται οι παρακάτω εντολές
    */
    if (SecondLink.Text == "Satyr Play")
    {
        TextTitleLabel.Text = "Euripides's Satyr Play"; //Αλλαγή
του Text του Label TextTitleLabel
        TextTitleLabel.Tag = "Title- Euripides's \"Satyr Play\"";
//Αλλαγή του Tag του Label TextTitleLabel
        TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
        FirstLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
        FirstLink.Text = "Tragedies"; //Αλλαγή του Text του
LinkLabel FirstLink
        FirstLink.Tag = "Link to Euripides \"Tragedies\" Text";
//Αλλαγή του Tag του LinkLabel FirstLink
        SecondLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
        SecondLink.Text = "Fragmentary Tragedies"; //Αλλαγή του
Text του LinkLabel FirstLink
        SecondLink.Tag = "Link to Euripides \"Fragmentary
Tragedies\" Text"; //Αλλαγή του Tag του LinkLabel FirstLink
        TragedyWritersText.Text = "1. Cyclops (uncertain date)";

```

```

        TragedyWritersText.Tag = "Tragedy Writers Text- Euripides
\"Satyr Play\""; //Αλλαγή του Tag του TextBox TragedyWritersText
    }
    else if (SecondLink.Text == "Fragmentary Tragedies")
    {
        TextTitleLabel.Text = "Euripides's Fragmentary Tragedies";
//Αλλαγή του Text του Label TextTitleLabel
        TextTitleLabel.Tag = "Title- Euripides's Fragmentary
Tragedies"; //Αλλαγή του Tag του Label TextTitleLabel
        TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
        FirstLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
        FirstLink.Text = "Tragedies"; //Αλλαγή του Text του
LinkLabel FirstLink
        FirstLink.Tag = "Link to Euripides \"Tragedies\" Text";
//Αλλαγή του Tag του LinkLabel FirstLink
        SecondLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
        SecondLink.Text = "Satyr Play"; //Αλλαγή του Text του
LinkLabel FirstLink
        SecondLink.Tag = "Link to Euripides \"Satyr Play\" Text";
//Αλλαγή του Tag του LinkLabel FirstLink
        TragedyWritersText.Text = "The following plays have come
down to us today only in fragmentary form; some consist of only a handful
of lines, but with some the fragments are extensive enough to allow
tentative reconstruction. \r\n1. Telephus (438 BC) \r\n2. Cretans (c. 435
BC) \r\n3. Stheneboea (before 429 BC) \r\n4. Bellerophon (c. 430 BC)
\r\n5. Cresphontes (ca. 425 BC) \r\n6. Erechtheus (422 BC) \r\n7. Phaethon
(c. 420 BC) \r\n8. Wise Melanippe (c. 420 BC) \r\n9. Alexandros (415 BC)
\r\n10. Palamedes (415 BC) \r\n11. Sisyphus (415 BC) \r\n12. Captive
Melanippe (412 BC) \r\n13. Andromeda (412 BC with Euripides' Helen)
\r\n14. Antiope (c. 410 BC) \r\n15. Archelaus (c. 410 BC) \r\n16.
Hypsipyle (c. 410 BC) r\n17. Philoctetes (c. 410 BC)"; //Αλλαγή του Text
στο TextBox TragedyWritersText
        TragedyWritersText.Tag = "Tragedy Writers Text-
Euripides's \"Fragmentary Tragedies\""; //Αλλαγή του Tag του TextBox
TragedyWritersText
    }
    else if (SecondLink.Text == "Fragmentary Plays")
    {
        TextTitleLabel.Text = "Sophocles's Fragmentary Plays";
//Αλλαγή του Text του Label TextTitleLabel
        TextTitleLabel.Tag = "Title- Sophocles's Work
\"Fragmentary Plays\""; //Αλλαγή του Tag του Label TextTitleLabel
        TextTitleLabel.Font = new Font("Arial", 10,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
        FirstLink.Visible = false; //Απόκρυψη του LinkLabel
FirstLink
        SecondLink.Visible = true; //Εμφάνιση του LinkLabel
FirstLink
        SecondLink.Text = "The Theban Plays"; //Αλλαγή του Text
του LinkLabel FirstLink
        SecondLink.Tag = "Link to Sophocles's \"The Theban Plays\"
Text"; //Αλλαγή του Tag του LinkLabel FirstLink
        TragedyWritersText.Text = "Fragments of The Tracking
Satyrs (Ichneutae) were discovered in Egypt in 1907. These amount to about
half of the play, making it the best preserved satyr play after Euripides'

```

```

Cyclops, which survives in its entirety. Fragments of The Progeny
(Epigonoi) were discovered in April 2005 by classicists at Oxford
University with the help of infrared technology previously used for
satellite imaging. The tragedy tells the story of the second siege of
Thebes. A number of other Sophoclean works have survived only in
fragments, including: \r\n• Aias Lokros (Ajax the Locrian) \r\n• Akhaiôn
Syllogos (The Gathering of the Achaeans) \r\n• Aleadae (The Sons of Aleus)
\r\n• Creusa \r\n• Eurypylus \r\n• Hermione \r\n• Inachos \r\n• Lacaenae
(Lacaenian Women) \r\n• Manteis or Polyidus (The Prophets or Polyidus)
\r\n• Nauplios Katapleon (Nauplius' Arrival) \r\n• Nauplios Pyrkaeus
(Nauplius' Fires) \r\n• Niobe \r\n• Oeneus \r\n• Oenomaus \r\n• Poimenes
(The Shepherds) \r\n• Polyxene \r\n• Syndeipnoi (The Diners, or, The
Banqueters) \r\n• Tereus \r\n• Thyestes \r\n• Troilus \r\n• Phaedra \r\n•
Triptolemus \r\n• Tyro Keiromene (Tyro Shorn) \r\n• Tyro Anagnorizomene
(Tyro Rediscovered)."; //Αλλαγή του Text στο TextBox TragedyWritersText
TragedyWritersText.Tag = "Tragedy Writers Text-
Sophocles's \"Fragmentary Plays\""; //Αλλαγή του Tag του TextBox
TragedyWritersText
}
else
{
    TextTitleLabel.Text = "Sophocles's Theban Plays";
//Αλλαγή του Text του Label TextTitleLabel
    TextTitleLabel.Font = new Font("Tahoma", 10,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
    TextTitleLabel.Tag = "Title- Sophocles's Work \"The Theban
Plays\""; //Αλλαγή του Tag του Label TextTitleLabel
    FirstLink.Visible = false; //Απόκρυψη του LinkLabel
    SecondLink.Visible = true; //Εμφάνιση του LinkLabel
    SecondLink.Text = "Fragmentary Plays"; //Αλλαγή του Text
του LinkLabel FirstLink
    SecondLink.Tag = "Link to Sophocles's\"Fragmentary Plays\"
Text"; //Αλλαγή του Tag του LinkLabel FirstLink
    TragedyWritersText.Text = "The Theban plays consist of
three plays: Antigone, Oedipus the King (also called Oedipus Tyrannus or
Oedipus Rex), and Oedipus at Colonus. All three plays concern the fate of
Thebes during and after the reign of King Oedipus. They have often been
published under a single cover. Sophocles, however, wrote the three plays
for separate festival competitions, many years apart. Not only are the
Theban plays not a true trilogy (three plays presented as a continuous
narrative) but they are not even an intentional series and contain some
inconsistencies among them. He also wrote other plays having to do with
Thebes, such as The Progeny, of which only fragments have survived. Other
than the three Theban plays, there are four surviving plays by Sophocles:
Ajax, The Trachiniae, Electra, and Philoctetes, the last of which won
first prize."; //Αλλαγή του Text στο TextBox TragedyWritersText
    TragedyWritersText.Tag = "Tragedy Writers Text-
Sophocles's \"The Theban Plays\" Works"; //Αλλαγή του Tag του TextBox
TragedyWritersText
}
}

private void CommentaryText_Click(object sender, EventArgs e)
//Μέθοδος για το event Click του MenuItem CommentaryText
{
    TragedyWritersText.Visible = true; //Εμφάνιση του TextBox
TragedyWritersText
}

```



```

        TextTitleLabel.Font = new Font("Tahoma", 11, FontStyle.Bold);
//Κατασκευή αντικειμένου Font για τη γραμματοσειρά του Label
TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά, μέγεθος
γραμματοσειράς, στυλ γραμματοσειράς
        ReadTheTextButton.Visible = false; //Απόκρυψη του Button
ReadTheTextButton
        HideTheTextButton.Visible = true; //Εμφάνιση του Button
HideTheTextButton
        increaseTextButton.Visible = true; //Εμφάνιση του button
increaseTextButton
        decreaseTextButton.Visible = true; //Εμφάνιση του button
decreaseTextButton

        /*
        * Χρήση εντολής λήψης απόφασης if για τον έλεγχο του property
Text του ComboBox TragedyWriterChoiceComboBox
        * Ανάλογα με την τιμή τύπου string που θα πάρει αυτό
        * θα πραγματοποιούνται οι παρακάτω εντολές
        * όταν το event Click του MenuItem CommentaryText συντελείται
        */
        if (TragedyWriterChoiceComboBox.Text == "Aeschylus")
//Περίπτωση όπου το ComboBox TragedyWriterChoiceComboBox παίρνει την τιμή
"Aeschylus"
        {
            TragedyWritersImage.Image = Properties.Resources.Aeshylus;
//Αλλαγή της Image στο PictureBox TragedyWritersImage
            TextTitleLabel.Text = "Commentary for Aeschylus";
//Αλλαγή του Text στο Label TextTitleLabel
            TextTitleLabel.Font = new Font("Tahoma", 11,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
            TextTitleLabel.Tag = "Text Title Label - Commentary for
Aeschylus Title"; //Αλλαγή του Tag στο Label TextTitleLabel
            TragedyWritersText.Text = "In the tone of his poems we
recognize the high-minded Athenian whose sword had drunk at Marathon and
Salamis the blood of long-haired Medes. His plays abound in military
terms, and while breathing the utmost contempt for barbarian prowess, he
often introduces on the stage the grotesque monsters whose images he had
seen among the spoils of Persian camps. Even his diction is of the
military type, his sonorous words striking on the ear like trumpet-sounds.
Yet he held his own dramas in light esteem, declaring that they were but
crumbs from the great banquet of Homer. \"Not only,\" says one of his
critics, \"had he fought at Marathon and Salamis against those Persians
whose rout he celebrated with patriotic pride, but he had been trained in
the Eleusinian mysteries, and was a passionate upholder of the institution
most intimately associated with the political traditions of the past- the
Areopagus. He had been born in the generation after Solon, to whose maxims
he fondly clung; he belonged to that anti-democratic party which favored
the Spartan alliance, and it was the Dorian development of Hellenic life
and the philosophical system based on it with which his religious and
moral convictions were imbued. Thus, even upon the generations which
succeeded him the chivalrous spirit and diction of his poetry, and the
unapproached sublimity of his dramatic imagination, fell, as it falls upon
later posterity, like the note of a mightier age.\" Such is the wonderful
power of art in its highest forms, that tragedy, as set forth by the three
great masters, and never treated with the same effect by their countless
imitators, has ever been held the gravest, most moral and most profitable
of all forms of poetry. \"Why is it,\" asks a classical scholar, \"that we
thrill with horror when the death- cry of Agamemnon announces the wreaking
of his doom? Why should we sympathize with Antigone as she sets forth on
her sacred mission- the burial of her brother- forbidden by man, but

```

enjoined by the command of Zeus? It is because in these and all other instances of the art of the Greek drama, while winged by the individual power of genius, is at the same time true to its purposes as an art, and in perfect harmony with Nature, who does not readily yield her secrets or teach her laws to all.\" Aeschylus was the only man of his age, or indeed of any age, who can compare with the great master of the modern drama in sublimity of conception and grandeur of poetic imagery. As to the esteem in which he was held by his contemporaries and his immediate posterity there is sufficient evidence, and first in the Frogs of Aristophanes, who there describes his temper as proud, stern and impatient; his sentiments as pure, noble and warlike; his genius inventive, magnificent and towering; his style lofty, bold and impetuous, full of gorgeous imagery and ponderous expressions, while in the dramatic arrangement of his pieces there remained much of ancient simplicity and somewhat of uncouth rudeness. Dionysius of Halicarnassus lauds the splendor of his talents, the propriety of his characters, the originality of his ideas and the force, variety and beauty of his language. Longinus speaks of the bold magnificence of his imagery, though condemning some of his conceptions as rude and turgid and his expressions as sometimes overstrained. Quintilian ascribes to him dignity of sentiment, sublimity of ideas and loftiness in style, yet often overcharged in diction and irregular in composition. Such, as seen through the eyes of antiquity, was the Shakespeare of the Greeks. "; //Αλλαγή του Text στο TextBox TragedyWritersText  
TragedyWritersText.Tag = "Tragedy Writers Text -  
Commentary for Aeschylus's Work"; //Αλλαγή του Tag του TextBox  
TragedyWritersText  
}  
else if (TragedyWriterChoiceCompoBox.Text == "Sophocles")  
//Περίπτωση όπου το ComboBox TragedyWriterChoiceCompoBox παίρνει την τιμή  
"Sophocles"  
{  
TragedyWritersImage.Image =  
Properties.Resources.Sophocles; //Αλλαγή της Image στο PictureBox  
TragedyWritersImage  
TextTitleLabel.Text = "Commentary for Sophocles";  
//Αλλαγή του Text στο Label TextTitleLabel  
TextTitleLabel.Font = new Font("Tahoma", 11,  
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του  
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,  
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς  
TextTitleLabel.Tag = "Text Title Label - Commentary for  
Sophocles Title"; //Αλλαγή του Tag στο Label TextTitleLabel  
TragedyWritersText.Text = "Aeschylus felt what a Greek  
tragedy ought to be as a religious union of the two elements of the  
national poetry. Sophocles, with his just perception of the beautiful in  
art, effected an outward realization of the conceptions of the great  
master, exhibiting in perfect form before the eyes of Athens what the  
other had hewn out in rude masses from the mines of thought. His tragedy  
was not essentially different from that of Aeschylus, and when he chose  
subjects which the latter had treated, his completed drama bore the same  
relation to its forerunner that a finished statue bears to an unfinished  
group. It was, as he thought, his mission to improve on the tragic art, as  
Phidias had improved the work of his predecessors. None did he deem worthy  
of the cothurnus save those who had figured in the ancient legends or in  
the poems of the epic cycle, and if an inferior character appears, it is  
only as the instrument of irony, introduced like a streak of bright color  
into the picture in contrast with its tragic gloom. Moreover,  
notwithstanding his sensualism, he was of a strongly religious  
temperament, filled with reverence for his country's gods, by whom, it  
would seem, he believed himself inspired. In the words which Landor aptly  
puts into his mouth, he declares himself to be \"only the interpreter of  
the heroes and divinities who are looking down upon him.\" An associate of

Pericles, though not one of his political disciples, Sophocles, in his full maturity stood, like the mighty ruler of the Greeks, amid a community to which both imparted the lustre of their genius on the sunny heights of noble and brilliant achievement, his perfect art typifying, as it were, the watchful and creative calm of his city's imperial epoch. Of a profoundly religious temperament, but without any vulgar superstition, he treats the sacred myths of his country in the spirit of a conscientious artist, contrasting, with many touches of irony, the struggles of humanity with the irresistible march of fate. After the retirement of Aeschylus, he was recognized as beyond dispute the greatest master of tragedy, and, as we have seen, during the lifetime of the former, wrested from him the tragic prize. The days of Sophocles were not altogether devoted to the muses. At the age of fifty-six he was appointed one of ten generals for the conduct of the war against Samos, but does not appear to have distinguished himself. Later he became a priest, and in extreme old age was elected one of a committee ordered, during the revolution brought about by Pisander, to investigate the condition of affairs and report thereon to the people. In the easy, good-natured way that was natural to him he assented to the establishment of an oligarchy under the council of four hundred as "a bad thing, but the least pernicious measure which circumstances allowed." In his last years the reverses of the Peloponnesian war, with their attendant civil dissensions, fell heavily on one whose chief delight was in domestic tranquility, and who still remembered the part which he bore in the glorious triumph of Salamis. Yet he was spared the misery of witnessing the final overthrow of his country, dying, full of years and honors, a few months before the defeat of Ægospotami wrought the downfall of Athens. Seven only of the dramas of Sophocles have come down to us, but these were, with one exception, composed in the full maturity of his tragic power, and each resplendent with its own peculiar excellencies. In the Antigone heroism is exhibited in a purely feminine character; in the Ajax, the manly sense of honor in all its strength. In the Trachiniæ, or Women of Trachis, are described the sufferings of Hercules and the levity of Déianeira, atoned for by her death; the Electra is distinguished by energy and pathos, and in the Oedipus at Colonus are a mildness and gracefulness suggestive of the character of the author. While we cannot divide the plays of Sophocles into distinct groups indicating certain periods in his dramatic art, he himself recognized three epochs in his own style- first, the tumid grandeur borrowed from Aeschylus; second, a harshness of expression due to his own mannerism; third, the style that seemed to him best fitted for the portrayal of human character.";

```
//Αλλαγή του Text στο TextBox
TragedyWritersText
    TragedyWritersText.Tag = "Tragedy Writers Text -
Commentary for Sophocles's Work"; //Αλλαγή του Tag του TextBox
TragedyWritersText
}
else //Περίπτωση όπου το ComboBox
TragedyWriterChoiceComboBox παίρνει οποιαδήποτε άλλη τιμή
{
    TragedyWritersImage.Image =
Properties.Resources.Euripides; //Αλλαγή της Image στο PictureBox
TragedyWritersImage
    TextTitleLabel.Text = "Commentary for Euripides";
//Αλλαγή του Text στο Label TextTitleLabel
    TextTitleLabel.Font = new Font("Tahoma", 11,
FontStyle.Bold); //Κατασκευή αντικειμένου Font για τη γραμματοσειρά του
Label TextTitleLabel με παραμέτρους (στην παρένθεση) γραμματοσειρά,
μέγεθος γραμματοσειράς, στυλ γραμματοσειράς
    TextTitleLabel.Tag = "Text Title Label - Commentary for
Euripides Title"; //Αλλαγή του Tag στο Label TextTitleLabel
    TragedyWritersText.Text = "Euripides focused on the
realism of his characters; for example, Euripides' Medea is a realistic
```

woman with recognizable emotions and is not simply a villain. In Hippolytus, Euripides writes in a particularly modern style, demonstrating how neither language nor sight aids in understanding in a civilization on its last leg. Euripides makes his point about vision both through the plot (Phaedra makes repeated references to her inability to see clearly and her wish to have her eyes covered), and through the sparseness of his staging, which lacked the dazzling elements that other plays often had. The same was true of his commentary on the use of language. The misuse of words played an important role in the storyline (Phaedra's letter, the nurse's betrayal of Phaedra's secret, Hippolytus' refusal to break his oath to save his own life, and his refusal to pay lip-service to Aphrodite), but in addition, the actual language of the play was often purposefully verbose and ungainly, again to show the ineffectual nature of language in comprehension in Euripides' age. According to Aristotle, Euripides' contemporary Sophocles said that he portrayed men as they ought to be, and Euripides portrayed them as they were. Euripides' realistic characterizations were sometimes at the expense of a realistic plot; Among the three extant ancient Greek tragedians, Euripides is particularly known for employing the literary device known as deus ex machina, whereby a god or goddess abruptly appears at drama's end to provide a contrived solution to an intractable problem. In the opinion of Aristotle, writing his Poetics a century later, this is an inadequate way to end a play. Many classicists cite this as a reason why Euripides was less popular in his own time.";

```

//Αλλαγή του Text στο TextBox TragedyWritersText
TragedyWritersText.Tag = "Tragedy Writers Text - Life of
Euripides"; //Αλλαγή του Tag του TextBox TragedyWritersText
    }
}

private void increaseTextButton_Click(object sender, EventArgs e)
{
    float k = TragedyWritersText.Font.Size; //Δήλωση μεταβλητή
για το Size της Font του TextBox TragedyWritersText, τύπου float
    if (k <= 18) //Εντολή απόφασης για τον έλεγχο της τιμής του
k
    {
        /* Κατασκευή αντικειμένου για το Font του TextBox
BasicInformationText.
        * Στη 2η παράμετρο (fontSize) γίνεται αύξηση κατά 1em στο
υπόρχον Size της Font του TextBox TragedyWritersText*/
        TragedyWritersText.Font = new Font("Tahoma", k + 1,
FontStyle.Regular);
    }
}

private void decreaseTextButton_Click(object sender, EventArgs e)
{
    float k = TragedyWritersText.Font.Size; //Δήλωση μεταβλητή
για το Size της Font του TextBox TragedyWritersText, τύπου float
    if (k >= 8) //Εντολή απόφασης για τον έλεγχο της τιμής του k
    {
        /* Κατασκευή αντικειμένου για το Font του TextBox
BasicInformationText.
        * Στη 2η παράμετρο (fontSize) γίνεται μείωση κατά 1em στο
υπόρχον Size της Font του TextBox TragedyWritersText*/
        TragedyWritersText.Font = new Font("Tahoma", k - 1,
FontStyle.Regular);
    }
}
}
}
}

```

## Κώδικας σχεδιασμού:

```
namespace Diploma_Work
{
    partial class TragedyWriters
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;
        private System.Windows.Forms.MainMenu mainMenu1;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
        disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.Windows.Forms.MenuItem OtherInfosMenu;
            System.ComponentModel.ResourceManager resources = new
System.ComponentModel.ResourceManager(typeof(TragedyWriters));
            this.CommentaryText = new System.Windows.Forms.MenuItem();
            this.WorksText = new System.Windows.Forms.MenuItem();
            this.WorkCategory1 = new System.Windows.Forms.MenuItem();
            this.WorkCategory2 = new System.Windows.Forms.MenuItem();
            this.WorkCategory3 = new System.Windows.Forms.MenuItem();
            this.LifeText = new System.Windows.Forms.MenuItem();
            this.mainMenu1 = new System.Windows.Forms.MainMenu();
            this.BackButton = new System.Windows.Forms.MenuItem();
            this.TragedyWritersImage = new
System.Windows.Forms.PictureBox();
            this.TragedyWriterChoiceCompoBox = new
System.Windows.Forms.ComboBox();
            this.TextTitleLabel = new System.Windows.Forms.Label();
            this.ReadTheTextButton = new System.Windows.Forms.Button();
            this.TragedyWritersText = new System.Windows.Forms.TextBox();
            this.HideTheTextButton = new System.Windows.Forms.Button();
            this.FirstLink = new System.Windows.Forms.LinkLabel();
            this.SecondLink = new System.Windows.Forms.LinkLabel();
            this.decreaseTextButton = new System.Windows.Forms.Button();
            this.increaseTextButton = new System.Windows.Forms.Button();
            OtherInfosMenu = new System.Windows.Forms.MenuItem();
            this.SuspendLayout();
            //
            // OtherInfosMenu
            //
        }
    }
}
```

```

        OtherInfosMenu.MenuItems.Add(this.CommentaryText);
        OtherInfosMenu.MenuItems.Add(this.WorksText);
        OtherInfosMenu.MenuItems.Add(this.LifeText);
        OtherInfosMenu.Text = "Other Infos";
        //
        // CommentaryText
        //
        this.CommentaryText.Text = "Commentary";
        this.CommentaryText.Click += new
System.EventHandler(this.CommentaryText_Click);
        //
        // WorksText
        //
        this.WorksText.MenuItems.Add(this.WorkCategory1);
        this.WorksText.MenuItems.Add(this.WorkCategory2);
        this.WorksText.MenuItems.Add(this.WorkCategory3);
        this.WorksText.Text = "Works";
        this.WorksText.Click += new
System.EventHandler(this.WorksText_Click);
        //
        // WorkCategory1
        //
        this.WorkCategory1.Text = "";
        this.WorkCategory1.Click += new
System.EventHandler(this.WorkCategory1_Click);
        //
        // WorkCategory2
        //
        this.WorkCategory2.Text = "";
        this.WorkCategory2.Click += new
System.EventHandler(this.WorkCategory2_Click);
        //
        // WorkCategory3
        //
        this.WorkCategory3.Text = "";
        this.WorkCategory3.Click += new
System.EventHandler(this.WorkCategory3_Click);
        //
        // LifeText
        //
        this.LifeText.Text = "Life";
        this.LifeText.Click += new
System.EventHandler(this.LifeText_Click);
        //
        // mainMenu1
        //
        this.mainMenu1.MenuItems.Add(OtherInfosMenu);
        this.mainMenu1.MenuItems.Add(this.BackButton);
        //
        // BackButton
        //
        this.BackButton.Text = "Back";
        this.BackButton.Click += new
System.EventHandler(this.BackButton_Click);
        //
        // TragedyWritersImage
        //
        this.TragedyWritersImage.Image =
((System.Drawing.Image)(resources.GetObject("TragedyWritersImage.Image")))
;
        this.TragedyWritersImage.Location = new
System.Drawing.Point(0, 23);

```

```

        this.TragedyWritersImage.Name = "TragedyWritersImage";
        this.TragedyWritersImage.Size = new System.Drawing.Size(240,
245);
        this.TragedyWritersImage.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        this.TragedyWritersImage.Tag = "The plague of";
        //
        // TragedyWriterChoiceCompoBox
        //
        this.TragedyWriterChoiceCompoBox.BackColor =
System.Drawing.Color.White;
        this.TragedyWriterChoiceCompoBox.Font = new
System.Drawing.Font("Tahoma", 10F, System.Drawing.FontStyle.Regular);
        this.TragedyWriterChoiceCompoBox.ForeColor =
System.Drawing.Color.Black;
        this.TragedyWriterChoiceCompoBox.Items.Add("Aeschylus");
        this.TragedyWriterChoiceCompoBox.Items.Add("Sophocles");
        this.TragedyWriterChoiceCompoBox.Items.Add("Euripides");
        this.TragedyWriterChoiceCompoBox.Location = new
System.Drawing.Point(12, 26);
        this.TragedyWriterChoiceCompoBox.Name =
"TragedyWriterChoiceCompoBox";
        this.TragedyWriterChoiceCompoBox.Size = new
System.Drawing.Size(217, 24);
        this.TragedyWriterChoiceCompoBox.TabIndex = 0;
        this.TragedyWriterChoiceCompoBox.Tag = "Tragedy writer
choosing list";
        this.TragedyWriterChoiceCompoBox.SelectedIndexChanged += new
System.EventHandler(this.TragedyWriterChoiceCompoBox_SelectedIndexChanged)
;
        //
        // TextTitleLabel
        //
        this.TextTitleLabel.Font = new System.Drawing.Font("Tahoma",
8F, System.Drawing.FontStyle.Regular);
        this.TextTitleLabel.Location = new System.Drawing.Point(0, 0);
        this.TextTitleLabel.Name = "TextTitleLabel";
        this.TextTitleLabel.Size = new System.Drawing.Size(240, 20);
        this.TextTitleLabel.Tag = "Text title label";
        this.TextTitleLabel.Text = "Choose a Tragedy Writer from the
list below";
        this.TextTitleLabel.TextAlign =
System.Drawing.ContentAlignment.TopCenter;
        //
        // ReadTheTextButton
        //
        this.ReadTheTextButton.Location = new System.Drawing.Point(12,
245);
        this.ReadTheTextButton.Name = "ReadTheTextButton";
        this.ReadTheTextButton.Size = new System.Drawing.Size(217,
20);
        this.ReadTheTextButton.TabIndex = 2;
        this.ReadTheTextButton.Tag = "Read the text button";
        this.ReadTheTextButton.Text = "Read the Text";
        this.ReadTheTextButton.Visible = false;
        this.ReadTheTextButton.Click += new
System.EventHandler(this.ReadTheTextButton_Click);
        //
        // TragedyWritersText
        //
        this.TragedyWritersText.BackColor =
System.Drawing.Color.White;

```



```

        this.TragedyWritersText.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
        this.TragedyWritersText.Location = new System.Drawing.Point(9,
23);
        this.TragedyWritersText.Multiline = true;
        this.TragedyWritersText.Name = "TragedyWritersText";
        this.TragedyWritersText.ReadOnly = true;
        this.TragedyWritersText.ScrollBars =
System.Windows.Forms.ScrollBars.Vertical;
        this.TragedyWritersText.Size = new System.Drawing.Size(222,
169);
        this.TragedyWritersText.TabIndex = 8;
        this.TragedyWritersText.Tag = "Tragedy writer text";
        //
        // HideTheTextButton
        //
        this.HideTheTextButton.Location = new System.Drawing.Point(7,
245);
        this.HideTheTextButton.Name = "HideTheTextButton";
        this.HideTheTextButton.Size = new System.Drawing.Size(222,
20);
        this.HideTheTextButton.TabIndex = 9;
        this.HideTheTextButton.Tag = "Hide the text button";
        this.HideTheTextButton.Text = "Hide the Text";
        this.HideTheTextButton.Click += new
System.EventHandler(this.HideTheTextButton_Click);
        //
        // FirstLink
        //
        this.FirstLink.Location = new System.Drawing.Point(9, 198);
        this.FirstLink.Name = "FirstLink";
        this.FirstLink.Size = new System.Drawing.Size(130, 20);
        this.FirstLink.TabIndex = 11;
        this.FirstLink.TabStop = false;
        this.FirstLink.Click += new
System.EventHandler(this.FirstLink_Click);
        //
        // SecondLink
        //
        this.SecondLink.Location = new System.Drawing.Point(9, 222);
        this.SecondLink.Name = "SecondLink";
        this.SecondLink.Size = new System.Drawing.Size(130, 20);
        this.SecondLink.TabIndex = 12;
        this.SecondLink.TabStop = false;
        this.SecondLink.Click += new
System.EventHandler(this.SecondLink_Click);
        //
        // decreaseTextButton
        //
        this.decreaseTextButton.BackColor =
System.Drawing.SystemColors.GrayText;
        this.decreaseTextButton.Font = new
System.Drawing.Font("Tahoma", 10F, System.Drawing.FontStyle.Bold);
        this.decreaseTextButton.ForeColor =
System.Drawing.Color.Black;
        this.decreaseTextButton.Location = new
System.Drawing.Point(201, 198);
        this.decreaseTextButton.Name = "decreaseTextButton";
        this.decreaseTextButton.Size = new System.Drawing.Size(28,
20);
        this.decreaseTextButton.TabIndex = 16;

```

```

        this.decreaseTextButton.Tag = "Button for decreasing the text
size";
        this.decreaseTextButton.Text = "-A";
        this.decreaseTextButton.Click += new
System.EventHandler(this.decreaseTextButton_Click);
        //
        // increaseTextButton
        //
        this.increaseTextButton.BackColor =
System.Drawing.SystemColors.GrayText;
        this.increaseTextButton.Font = new
System.Drawing.Font("Tahoma", 10F, System.Drawing.FontStyle.Bold);
        this.increaseTextButton.ForeColor =
System.Drawing.Color.Black;
        this.increaseTextButton.Location = new
System.Drawing.Point(167, 198);
        this.increaseTextButton.Name = "increaseTextButton";
        this.increaseTextButton.Size = new System.Drawing.Size(28,
20);
        this.increaseTextButton.TabIndex = 15;
        this.increaseTextButton.Tag = "Button for increasing the text
size";
        this.increaseTextButton.Text = "+A";
        this.increaseTextButton.Click += new
System.EventHandler(this.increaseTextButton_Click);
        //
        // TragedyWriters
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(96F, 96F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Dpi;
        this.AutoScroll = true;
        this.ClientSize = new System.Drawing.Size(240, 268);
        this.Controls.Add(this.decreaseTextButton);
        this.Controls.Add(this.increaseTextButton);
        this.Controls.Add(this.SecondLink);
        this.Controls.Add(this.FirstLink);
        this.Controls.Add(this.TextTitleLabel);
        this.Controls.Add(this.HideTheTextButton);
        this.Controls.Add(this.ReadTheTextButton);
        this.Controls.Add(this.TragedyWritersText);
        this.Controls.Add(this.TragedyWriterChoiceCompoBox);
        this.Controls.Add(this.TragedyWritersImage);
        this.Menu = this.mainMenu1;
        this.Name = "TragedyWriters";
        this.Tag = "Tragedy writers";
        this.Text = "Tragedy Writers";
        this.Activated += new
System.EventHandler(this.TragedyWriters_Activated);
        this.ResumeLayout(false);
    }

#endregion

private System.Windows.Forms.PictureBox TragedyWritersImage;
private System.Windows.Forms.Label TextTitleLabel;
private System.Windows.Forms.MenuItem BackButton;
private System.Windows.Forms.ComboBox TragedyWriterChoiceCompoBox;
private System.Windows.Forms.Button ReadTheTextButton;
private System.Windows.Forms.TextBox TragedyWritersText;
private System.Windows.Forms.Button HideTheTextButton;
private System.Windows.Forms.MenuItem LifeText;

```

```
private System.Windows.Forms.MenuItem WorksText;  
private System.Windows.Forms.MenuItem CommentaryText;  
private System.Windows.Forms.MenuItem WorkCategory1;  
private System.Windows.Forms.MenuItem WorkCategory2;  
private System.Windows.Forms.MenuItem WorkCategory3;  
private System.Windows.Forms.LinkLabel FirstLink;  
private System.Windows.Forms.LinkLabel SecondLink;  
private System.Windows.Forms.Button decreaseTextButton;  
private System.Windows.Forms.Button increaseTextButton;  
}  
}
```

# Βιβλιογραφία

---

## Βιβλία- Εκθέσεις

- [1]. Commission of the European Communities (2005) *COMMUNICATION FROM THE COMMISSION TO THE COUNCIL, THE EUROPEAN PARLIAMENT AND THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS eAccessibility*. [SEC(2005)1095], Brussels: 13.9.2005COM(2005)425 final.
- [2]. Commission of the European Communities (2002) *COMMUNICATION FROM THE COMMISSION TO THE COUNCIL, THE EUROPEAN PARLIAMENT AND THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS, eEurope: Accessibility of Public Web Sites and their Content*, Brussels: 25.09.2001COM(2001) 529 final.
- [3]. Ferguson Jeff, Patterson Brian, Beres Jason, Boutquin Pierre, Gupta Meeta (2002), *C# Bible*. Indianapolis, Indiana: Wiley Publishing, Inc.
- [4]. Greanier Todd (2004), *Java, Εισαγωγή στη Σύγχρονη Τεχνολογία*. 2<sup>η</sup> Ελληνική Έκδοση: Μ. Γκιούρδας.
- [5]. Liberty Jesse and Xie Donald (2008), *Programming C# 3.0*. 5<sup>th</sup> Edition Gravenstein Highway North, Sebastopol: O'Reilly.
- [6]. Nielsen Jakob (1993), *Usability Engineering*. San Francisco: Morgan Kaufmann.
- [7]. Powers Lars, Snell Mike (2008), *Microsoft Visual Studio 2008 Unleashed*. Indianapolis, Indiana: Pearson Education, Inc.
- [8]. Randolph Nick, Gardner David (2008), *Professional Visual Studio 2008*. Indianapolis, Indiana: Wiley Publishing, Inc.
- [9]. Robinson Simon, Nagel Christian, Watson Karli, Glynn Jay, Skinner Morgan, Evjen Bill (2004), *Professional C#*. 3<sup>rd</sup> Edition Indianapolis, Indiana: Wiley Publishing, Inc.
- [10]. Schildt Herbert (2009), *C# 3.0- A Beginner's Guide*. MacGraw.
- [11]. Sharp John (2008), *Microsoft Visual C# 2008 Step by Step*. Redmont, Washington: Microsoft Press.
- [12]. Γιακουμίδης Μανόλης, Διαμαντίδης Νίκος (2009), *Τεχνολογία Λογισμικού*. Αθήνα: Εκδόσεις Αθ. Σταμούλης.

## Ιστοσελίδες

- [13]. Europa: Getaway to the European Union *Web Accessibility Policy*. Προσβάσιμο στη θέση: [http://europa.eu/geninfo/accessibility\\_policy\\_en.htm](http://europa.eu/geninfo/accessibility_policy_en.htm) (τελευταία επίσκεψη: 22 Σεπτεμβρίου 2010).
- [14]. Gartner (2010) *Gartner Newsroom, Press Releases: Gartner Says Worldwide Mobile Device Sales Grew 13.8 Percent in Second Quarter of 2010, But Competition Drove Prices Down*. Garner, Inc. Προσβάσιμο στη θέση: <http://www.gartner.com/it/page.jsp?id=1421013> (τελευταία επίσκεψη: 12 Οκτωβρίου 2010)
- [15]. Microsoft (2010) *Microsoft Accessibility Technology for Everyone*. Προσβάσιμο στη θέση: <http://www.microsoft.com/enable/demos/windows7/acctoday.aspx> (τελευταία επίσκεψη: 22 Σεπτεμβρίου 2010).
- [16]. MSDN (2010) *Windows Mobile Development Center*. Microsoft Corporation. Προσβάσιμο στη θέση: <http://msdn.microsoft.com/en-us/windowsmobile/default.aspx> (τελευταία επίσκεψη: 12 Οκτωβρίου 2010)
- [17]. Section508.gov (2010) *Opening doors to IT*. Προσβάσιμο στη θέση: <http://www.section508.gov/index.cfm> (τελευταία επίσκεψη: 22 Σεπτεμβρίου 2010).
- [18]. Stephanidis, C. (1998) "Toward an Information Society for All: An International R&D Agenda", *International Journal of Human-Computer Interaction* , 10 (2), pp. 107-113. [Online] DOI: [http://www.ics.forth.gr/proj/at-hci/files/white\\_paper\\_1998.pdf](http://www.ics.forth.gr/proj/at-hci/files/white_paper_1998.pdf) (τελευταία επίσκεψη: 22 Σεπτεμβρίου 2010).
- [19]. WAI (2009) *Web Accessibility Initiative*. W3C. Προσβάσιμο στη θέση: <http://www.w3.org/WAI> (τελευταία επίσκεψη: 15 Οκτωβρίου 2010).
- [20]. WCAG (2008) *Web Content Accessibility Guidelines*. W3C. Προσβάσιμο στη θέση: <http://www.w3.org/TR/WCAG20/> (τελευταία επίσκεψη: 15 Οκτώβρη 2010).
- [21]. Wikipedia: The free Encyclopedia (2010). Προσβάσιμο στη θέση: [http://en.wikipedia.org/wiki/Main\\_Page](http://en.wikipedia.org/wiki/Main_Page) (τελευταία επίσκεψη: 22 Σεπτεμβρίου 2010).
- [22]. Εύτεχνος (2004- 2005) *Υποστηρικτικές Τεχνολογίες και Προσβασιμότητα στο Διαδίκτυο για άτομα με αναπηρίες (ΑμεΑ)*. Προσβάσιμο στη θέση: <http://e-bility.gr/eutexnos/default.asp> (τελευταία επίσκεψη: 22 Σεπτεμβρίου 2010).
- [23]. 100 People: A World Portrait. Προσβάσιμο στη θέση: [http://www.100people.org/statistics\\_100stats.php?section=statistics](http://www.100people.org/statistics_100stats.php?section=statistics) (τελευταία επίσκεψη: 22 Σεπτεμβρίου 2010).

