



ΤΕΙ ΜΕΣΣΟΛΟΓΓΙΟΥ
ΠΑΡΑΡΤΗΜΑ ΝΑΥΠΑΚΤΟΥ
ΤΜΗΜΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΔΙΚΤΥΩΝ



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ
ΕΥΡΩΠΑΪΚΗ ΕΚΚΣΗ
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
ΕΥΡΩΠΑΪΚΟ ΤΑΜΕΙΟ ΠΕΡΙΦΕΡΕΙΑΚΗΣ ΑΝΑΠΤΥΞΗΣ



ΠΑΙΔΕΙΑ ΜΠΡΟΣΤΑ
2^ο Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης

Ανάπτυξη Εκπαιδευτικού Λογισμικού σε Macromedia Flash για το μάθημα των Δικτύων

Πτυχιακή εργασία

Μανουράς Ιάσωνας

Επιβλέπων: Χριστοδούλου Σωτήριος

Οκτώβριος 2012

Ναύπακτος

Περίληψη

Οι υπολογιστές αποτελούν πλέον μία πραγματικότητα στην καθημερινότητα των ανθρώπων, τόσο για σκοπούς διασκέδασης όσο και για εργασιακούς σκοπούς. Καθώς οι ανάγκες στην εκπαίδευση ολοένα αυξάνονται σύμφωνα με τα καινούργια τεχνολογικά επιτεύγματα, το εκπαιδευτικό λογισμικό πλέον έχει εδραιωθεί ως ένας εναλλακτικός αλλά και αποδοτικός τρόπος εκπαίδευσης και διδασκαλίας. Σκοπός της πτυχιακής αυτής είναι να συγκεντρώσει και να παρουσιάσει τις θεωρίες, τα χαρακτηριστικά και τις ανάγκες που εκπληρώνουν τα λογισμικά αυτά, και με βάση αυτά το σχεδιασμό και την ανάπτυξη ενός εκπαιδευτικού λογισμικού για το μάθημα των Δικτύων. Παρουσιάζουμε την έρευνά μας στον τομέα των εκπαιδευτικών λογισμικών, με έμφαση στη διαδραστικότητα και την επικοινωνία ανθρώπου-υπολογιστή, καθώς και τη διαδικασία που ακολουθήσαμε για την ανάπτυξη του εκπαιδευτικού λογισμικού σύμφωνα με τις προδιαγραφές που προστάσουν οι σύγχρονες έρευνες. Τα εκπαιδευτικά λογισμικά που παρουσιάζουμε χαρακτηρίζεται από διαδραστικότητα με το χρήστη, πολυμεσικό υλικό, διαδραστικές εφαρμογές όπως παιχνίδια, ενώ παράλληλα λαμβάνουμε υπόψιν τυχόν μαθησιακές δυσκολίες του χρήστη (όπως για παράδειγμα δυσκολία εκμάθησης διδακτικής ύλης από κείμενο), και το γνωστικό υπόβαθρο του χρήστη στο αντικείμενο των Δικτύων.

Περιεχόμενα

Κεφάλαιο 1. Εισαγωγή

Κεφάλαιο 2. Βιβλιογραφική Ανασκόπηση για Εκπαιδευτικά Λογισμικά.

2.1 Μικτά μαθησιακά Περιβάλλοντα ---Blended learning Environments.

2.2 Ισχυρά Εκπαιδευτικά Περιβάλλοντα

2.3.1 Διαδραστικά Εκπαιδευτικά Περιβάλλοντα

2.3.2 Επικοινωνία και Εκπαίδευση

2.3.3 Πολυμέσα και Εκπαίδευση

2.4 Παιχνίδια και Εκπαίδευση

2.5 Τηλεπικοινωνίες και Εκπαίδευση.

2.6 Εξ' Αποστάσεως Εκπαίδευση --Distance learning

2.7.1 Χαρακτηριστικά Εκπαιδευτικών Λογισμικών

2.7.2 Αλληλεπιδραστικά Εκπαιδευτικά Περιβάλλοντα

2.7.3 Αξιολόγηση Εκπαιδευτικού Λογισμικού

2.8 Συμπέρασμα

Κεφάλαιο 3. Επιλέγοντας την Κατάλληλη Τεχνολογία

3.1 Εισαγωγή

3.2.1 Τεχνολογία Flash

3.2.2 Χαρακτηριστικά του Flash

3.2.3 Προτερήματα του Flash

3.2.4 Αδυναμίες του Flash

3.2.5 Το μέλλον του Flash

3.3. Συμπέρασμα

Κεφάλαιο 4. Προγραμματιστικές Γλώσσες

4.1. Εισαγωγή

4.2.1 Action Script

4.2.2 Κλάσεις και Αντικείμενα

4.3 HTML

4.4 XML

4.5 CSS

Κεφάλαιο 5. Σχεδιασμός Εκπαιδευτικού Λογισμικού

5.1 Εισαγωγή

5.2. Περίληψη του Εκπαιδευτικού Λογισμικού

5.3.1 Χαρακτηριστικά Διδαχθείσας Ύλης

5.3.2 Πολυμεσικό Υλικό

5.3.3 Παιχνίδια

5.3.4 Τεστ

5.3.5 Κουίζ

5.4 Επικοινωνία

5.5 Χαρακτηριστικά Λογισμικού

Κεφάλαιο 6. Υλοποίηση Εκπαιδευτικού Λογισμικού

6.1. Εισαγωγή

6.2. Υπό-εφαρμογές

6.3.1 Intro

6.3.2 Ανάπτυξη

6.4.1 Τεστ

6.4.2 Ανάπτυξη

6.5.1 Κρεμάλα

6.5.2 Ανάπτυξη

6.6.1 Quiz

6.6.2 Ανάπτυξη

6.7 Βασικός Σκελετός Εφαρμογής

6.8.1 Απλή σκηνή

6.8.2 Κουμπιά

6.8.3 Επιλογή Γνωστικού Επιπέδου

6.9.1 Ενότητες Μενού

6.9.2 Κυρίως Σκελετός

Κεφάλαιο 7. Συμπεράσματα

7.1 Μελλοντική Δουλειά

Κώδικας

Βιβλιογραφία

Κεφάλαιο 1. Εισαγωγή

Η διάσταση της χρήσεως των υπολογιστών, και ιδιαιτέρως του Internet, είναι μεγαλύτερη από ποτέ. Η ραγδαία διάδοση του Internet σε συνεργασία με την εύκολη πρόσβαση σε αυτό, έχουν δημιουργήσει μια νέα εποχή και στην εκπαίδευση. Στις μέρες μας, η διαδικασία της διδασκαλίας και της εκπαίδευσης κρίνεται αναγκαία να συνδεθεί με τις τεχνολογικές αυτές αλλαγές, καθώς και να προσφέρει ευκολότερες διόδους πρόσβασης στην εκπαίδευση.

Οι πιθανές ευεργετικές ιδιότητες των υπολογιστών στη διαδικασία της διδασκαλίας, έγιναν αντιληφθείς από πολύ νωρίς. Οι υπολογιστές όχι μόνο μετατρέπουν τη διδασκαλία σε μια πιο ενδιαφέρον διαδικασία για τον εκπαιδευόμενο, αλλά κρύβουν και οικονομικά οφέλη (Woodhouse and McDougall, 1986). Αυτό οδήγησε στην ανάπτυξη λογισμικών σχεδιασμένα για τη διδασκαλία, συνήθως κάνοντας χρήση διδακτικών θεωριών, και μαθησιακής ύλης από υπάρχον εκπαιδευτικά ιδρύματα. Παρόλο το γεγονός ότι οι υπολογιστές, ακόμα, δεν είναι δυνατόν να αντικαταστήσουν έναν εκπαιδευτικό, η μεγάλη πρόοδος στον τομέα των εκπαιδευτικών λογισμικών, σε συνδυασμό με τις εξελίξεις στον τομέα διαδραστικότητας-ανθρώπου-υπολογιστή έχουν εδραιώσει πλέον τα εκπαιδευτικά λογισμικά στον τομέα της εκπαίδευσης.

Η ζήτηση για εκπαιδευτικά λογισμικά αυξάνεται εκθετικά με την αύξηση του ενδιαφέροντος για την εκπαιδευτική μεταρρύθμιση, την ενσωμάτωση της τεχνολογίας στην εκπαίδευση, το Internet, και την εξ'αποστάσεως εκπαίδευση. Οι εκπαιδευτικές εφαρμογές πρέπει να είναι ευέλικτες διότι τα προγράμματα σπουδών και οι μέθοδοι διδασκαλίας ποικίλλουν και διαφέρουν σημαντικά μεταξύ τους. Για την ικανοποίηση αυτών των αναγκών ένα ευρύ φάσμα από πανεπιστήμια, προγραμματιστές και εταιρίες αναπτύσσουν το αποκαλούμενο

εκπαιδευτικό λογισμικό, το οποίο κατηγοριοποιείται ανάλογα με τις ανάγκες που καλύπτει, τις προγραμματιστικές μεθόδους που χρησιμοποιεί, αλλά και τις διδακτικές τακτικές που ακολουθεί.

Η πτυχιακή εργασία που παρουσιάζουμε, ασχολείται με το σχεδιασμό και την ανάπτυξη ενός εκπαιδευτικού λογισμικού σε Flash, για το μάθημα των Δικτύων. Στο επερχόμενο κεφάλαιο, Κεφάλαιο 2, παρουσιάζουμε τα αποτελέσματα της έρευνάς μας σχετικά με τις υπάρχουσες θεωρίες διδασκαλίας για τα εκπαιδευτικά λογισμικά, καθώς και τα τεχνολογικά επιτεύγματα που έχουν αναφερθεί στον τομέα αυτόν. Προφανώς, δεν θα ήταν εφικτό να παρουσιάσουμε όλη τη δουλειά που έχει γίνει στον κλάδο αυτό, καθότι τεράστια. Για τον λόγο αυτό, αποφασίσαμε να παρουσιάσουμε μόνο τις δουλειές και τις έρευνες στις οποίες βασιστήκαμε για την ανάπτυξη του δικού μας εκπαιδευτικού λογισμικού, για το μάθημα των Δικτύων. Στη συνέχεια, Κεφάλαιο 3, παρουσιάζουμε τον λόγο που επιλέξαμε την τεχνολογία Flash για την ανάπτυξη του λογισμικού μας. Έπειτα, στο Κεφάλαιο 4, παρουσιάζουμε τις προγραμματιστικές γλώσσες που χρησιμοποιήσαμε κατά την ανάπτυξη του εκπαιδευτικού λογισμικού. Στο Κεφάλαιο 5 παρουσιάζουμε τις θεωρίες που βασιστήκαμε, και τη διαδικασία σχεδιασμού που ακολουθήσαμε προκειμένου να δημιουργήσουμε ένα αποδοτικό και παράλληλα ενδιαφέρον εκπαιδευτικό λογισμικό, για τις ανάγκες του χρήστη. Στο Κεφάλαιο 6 παρουσιάζουμε μια εκτενής ανάλυση της διαδικασίας ανάπτυξης που ακολουθήσαμε, τόσο για το γραφικό περιβάλλον όσο για την προγραμματιστική διαδικασία - τα προγραμματιστικά κόλπα και τις προγραμματιστικές δυσκολίες που συναντήσαμε. Τέλος, στο Κεφάλαιο 7 αναπτύσσουμε τα συμπεράσματά μας, καθώς και κάποιες πιθανές μελλοντικές βελτιώσεις του λογισμικού μας.

Κεφάλαιο 2. Βιβλιογραφική Ανασκόπηση για Εκπαιδευτικά Λογισμικά.

2.1 Μικτά μαθησιακά Περιβάλλοντα ---Blended learning Environments.

Ο όρος blended η αλλιώς μικτή εκμάθηση, στον τομέα της εκπαιδευτικής έρευνας αναφέρεται σε μια μίξη διαφορετικών περιβαλλόντων μάθησης. Συνδυάζει την παραδοσιακή μέθοδο πρόσωπο---με---πρόσωπο (κοινώς διδασκαλία εντός τάξης), σε συνδυασμό με πιο σύγχρονους μεθόδους κατά κόρον βασιζόμενες στη χρήση του υπολογιστή. Σύμφωνα με τους υπερασπιστές της, η στρατηγική αυτή δημιουργεί μια πιο ολοκληρωμένη προσέγγιση τόσο για τους εκπαιδευτές και τους εκπαιδευόμενους. Σε άλλες περιπτώσεις, μεγαλύτερη εξάρτηση από την τεχνολογία μέσα στην τάξη μπορεί να συμβεί. Οι δραστηριότητες μπορούν να διαρθρώνεται γύρω από την πρόσβαση σε ηλεκτρονικές πηγές, η επικοινωνία μέσω των κοινωνικών μέσων μαζικής ενημέρωσης ή αλληλεπίδραση με τους μαθητές εξ αποστάσεως και σε άλλες τάξεις ή άλλα μαθησιακά περιβάλλοντα.

Υπάρχουν πολλές διαφορετικές προσεγγίσεις με τη σύνθετη μάθηση. Μπορεί να πάρει πολλές μορφές ανάλογα με τους εκπαιδευτικούς και τους μαθητές που συμμετέχουν. Όσο για σήμερα, δεν υπάρχει συναίνεση σχετικά με ένα ενιαίο συμφωνημένο ορισμό για μικτή μάθηση. Οι όροι “μικτή”, “υβριδικό” και “μικτής λειτουργίας” χρησιμοποιούνται εναλλακτικά στην τρέχουσα βιβλιογραφία της έρευνας

2.2 Ισχυρά Εκπαιδευτικά Περιβάλλοντα

Ο (Smeets, 2005) περιγράφει ως ισχυρά εκπαιδευτικά περιβάλλοντα εκείνα τα οποία ενθαρρύνουν τις βέλτιστες διαδικασίες εκμάθησης. Σε τέτοια περιβάλλοντα οι μαθητές συμμετέχουν ενεργά (άλλοτε ανεξάρτητα και άλλοτε σε συνεργασία) για την διεκπεραίωση αυθεντικών, πλούσιων εργασιών που έχουν προσαρμοστεί στις ατομικές τους ανάγκες. Αυτά τα χαρακτηριστικά, για τη δημιουργία ισχυρών εκπαιδευτικών περιβαλλόντων, συμπεριλαμβάνονται στη δημοσίευση των Christie και Boyd (2007) η οποία αφορά τα μηνύματα κλειδιά από την ερευνητική βιβλιογραφία για τη δημιουργία ενός προγράμματος σπουδών με στόχο την τελειότητα. Η ενεργός συμμετοχή θεωρείται ως ένας από τους βασικούς στόχους κατά τη σχεδίαση της κατάλληλης διδασκαλίας και τις μαθησιακές προσεγγίσεις. Κατά το σχεδιασμό της εκμάθησης των φοιτητών εκτός εκπαιδευτικών κέντρων θεωρείται απαραίτητο χαρακτηριστικό η ανεξαρτησία αλλά και η συνεργασία.

2.3.1 Διαδραστικά Εκπαιδευτικά Περιβάλλοντα

Στη δουλειά (Schaffer και Hannaffin, 1993) υποδεικνύεται ότι η αυξημένη διαδραστικότητα κατά τη διάρκεια μίας εκπαιδευτικής δραστηριότητας, φέρει σαν αποτέλεσμα την αύξηση των κινήτρων των μαθητών για τη συμμετοχή τους καθώς και τη δημιουργία μίας πιο θετικής αντιμετώπισης των μαθητών ως προς την εκπαιδευτική αυτή δραστηριότητα.

Η διαδραστικότητα κα τη διάρκεια μίας εκπαιδευτικής δραστηριότητας μπορεί να λάβει διάφορες μορφές, εξαρτώντας από το βαθμό της συμμετοχής των εκπαιδευόμενων. Για παράδειγμα ως διαδραστική δραστηριότητα μπορεί να θεωρηθεί η απάντηση σε ένα

ερώτημα, η επιλογή της ύλης την οποία θέλει να διδαχθεί, όπως και η επικοινωνία με τον εκπαιδευτή. Οι Markwood και Johnson (1994), κατηγοριοποιούν τις διαδραστικές δραστηριότητες σε:

- ⤴ Διαδραστικότητα υπό την έννοια της χρήσης πολυμεσικών μέσων όπως είναι το βίντεο, ο ήχος και οι εικόνες.
- ⤴ Διαδραστικότητα υπό τη μορφή επικοινωνίας με τον εκπαιδευτή ή με άλλους εκπαιδευόμενους.
- ⤴ Διαδραστικότητα υπό την έννοια συνεργασίας πολλών εκπαιδευόμενων.
- ⤴ Διαδραστικότητα υπό την μορφή αποτελεσμάτων: οι κινήσεις και οι αποφάσεις του εκπαιδευόμενου αποφέρουν κάποια αποτελέσματα, για παράδειγμα κάποιο σκορ.

2.3.2 Επικοινωνία και Εκπαίδευση

Ένα κοινό στοιχείο της εκπαίδευσης που λαμβάνει χώρα μέσα σε τάξεις, είναι η επικοινωνία, η κοινωνικοποίηση και η διαδραστικότητα του εκπαιδευόμενου με τον εκπαιδευτή (Stubbs, 1976). Η διαδικασία κα την οποία ο εκπαιδευόμενος μοιράζεται κάποια άποψη, δέχεται ερωτήσεις, ή μοιράζεται κάποια διαφωνία θεωρείται άκρως σημαντικό συστατικό για μία επιτυχημένη διαδικασία εκμάθησης.

Τα εκπαιδευτικά λογισμικά, τα οποία κυρίως αποτελούν ένα μέσω υπολογιστή-με-άνθρωπο εκπαίδευσης, πρέπει να λάβουν υπόψιν τα διαδραστικά συστατικά της εκπαίδευσης που λαμβάνει χώρα μέσα σε μία τάξη, και να τα προσομοιώσουν με τον καλύτερο δυνατό τρόπο. Για παράδειγμα, ένα αποτελεσματικό εκπαιδευτικό λογισμικό θα πρέπει να λαμβάνει υπόψιν

του την ανάγκη του εκπαιδευόμενου για επικοινωνία με τον εκπαιδευτή –είτε για να εκφράσει κάποιες απορίες ή απόψεις, είτε απλά για να εκπληρώσει τις κοινωνικές ανάγκες του με την επαφή με τον καθηγητή. Κάτι τέτοιο, μπορεί να επιτευχθεί με ποικίλους τρόπους: από τη δημιουργία μίας απλής φόρμας επικοινωνίας με τον εκπαιδευτή, μέσω της οποίας ο εκπαιδευόμενος μπορεί να εκφράσει τις απόψεις του, ανησυχίες και ερωτήσεις. Ένας πιο προηγμένος τρόπος, αποτελεί η δημιουργία περιβαλλόντων τα οποία προσφέρουν και εικονικούς καθηγητές ικανούς να συμμετέχουν σε διαδραστικές δραστηριότητες με τον εκπαιδευόμενο.

2.3.3 Πολυμέσα και Εκπαίδευση

Κατά την περίοδο 1980 με 1990 η έννοια των πολυμέσων πήρε μία καινούρια διάσταση, καθώς παρουσιάστηκαν τεχνικές σχεδίασης πολυμέσων με εκπληκτικές δυνατότητες. Σε συνδυασμό με τις εκάστοτε ικανότητες του υλικού και του λογισμικού των υπολογιστών παράχθηκαν τεχνολογίες για την ενίσχυση της διαδικασίας της εκπαίδευσης οι οποίες μπορούν να λάβουν υπόψιν τις ανάγκες των εκπαιδευόμενων.

Μία βασική διαδραστική εφαρμογή βασισμένη σε πολυμέσα , παρέχει στον χρήστη τον πλήρη έλεγχο έτσι ώστε να ξεναγηθεί στο παρεχόμενο ο υλικό με την ταχύτητα που επιθυμεί και να πλοηγηθεί βασισμένος στα προσωπικά του ενδιαφέροντα και ανάγκες. Ο βασικός στόχος των διαδραστικών πολυμεσικών εφαρμογών δεν είναι η αντικατάσταση του δασκάλου/δασκάλας αλλά ο εμπλουτισμός του ρόλου αυτού.

Πέραν του ότι είναι ένα ισχυρό εργαλείο για τη δημιουργία παρουσιάσεων, τα πολυμέσα προσφέρουν μοναδικά πλεονεκτήματα στον τομέα της εκπαίδευσης. Για παράδειγμα, ένα

γυμνό κείμενο δεν επιτρέπει τους φοιτητές λογοτεχνίας να πάρουν μια αίσθηση από οποιοδήποτε από τα έργα του Σαίξπηρ. Τα πολυμέσα μετατρέπουν την απλή διδασκαλία σε μία πιο ζωντανή εμπειρία. Το κλειδί είναι η ταυτόχρονη μετάδοση γραφικών, βίντεο και ήχου πέραν ενός απλού κειμένου. Τα πολυμέσα δίνουν τη δυνατότητα εκμάθησης μέσω εξερεύνηση, την ανακάλυψη και την εμπειρία.

Η τεχνολογία δεν οδηγεί κατ' ανάγκη την εκπαίδευση. Ο ρόλος αυτός ανήκει στις μαθησιακές ανάγκες των εκπαιδευόμενων. Μέσω των πολυμέσων, η διαδικασία της μάθησης μπορεί να γίνει περισσότερο προσανατολισμένη ως προς το στόχο, πιο διαδραστική με τον εκπαιδευόμενο. Τα πολυμέσα μπορούν να δώσουν τη δυνατότητα για παροχή διασκεδαστικής και φιλικής εκπαίδευσης, αφαιρώντας όποιον φόβο ανεπάρκειας ικανοτήτων ή αποτυχίας.

Η παιδαγωγική δύναμη των πολυμέσων είναι ότι χρησιμοποιεί το φυσικό τρόπο για επεξεργασία πληροφορίας χρησιμοποιώντας τις αισθήσεις του εκπαιδευόμενου ως αποδέκτη. Η όραση και η ακοή σε συνδυασμό με τη σκέψη αποτελούν ένα εξαιρετικό σύστημα για τη μετατροπή των δεδομένων μέσα από τις αισθήσεις σε πληροφορίες. «Μια εικόνα αξίζει όσο χίλιες λέξεις»

Για το μαθητή, ένα πλεονέκτημα των πολυμέσων ως προς την εκπαίδευση είναι η ποικιλία που προσφέρεται. Έστω και οι ελάχιστες δυνατές εικόνες που μπορεί να προσφέρει ένα εκπαιδευτικό λογισμικό, δημιουργούν ανακούφιση στον εκπαιδευόμενο σε σχέση με ένα εκπαιδευτικό λογισμικό το οποίο είναι ολόκληρο βασισμένο μόνο πάνω σε κείμενο.

Εμπλουτισμό στο εκπαιδευτικό λογισμικό όμως δεν προσφέρουν μόνο οι εικόνες, αλλά οποιοδήποτε πολυμεσικό στοιχείο δεν είναι κείμενο, για παράδειγμα ήχος, βίντεο κ.λπ..

Παρόλα αυτά η χρήση των πολυμέσων μερικές φορές αντιμετωπίζει κάποια προβλήματα. Ένα από τα αρνητικά των πολυμέσων είναι ότι επί το πλείστον απαιτούν συγκεκριμένα λογισμικά αναπαραγωγής των πολυμέσων αν όχι και high-end συστήματα υπολογιστών. Η αναπαραγωγή ήχου, εικόνων, κινουμένων σχεδίων, και ειδικά βίντεο, απαιτούν την ανταλλαγή μεγάλων ποσοτήτων δεδομένων. Για τον λόγο αυτό χρειάζονται η αναπαραγωγή πολυμέσων επιβάλλει τη χρήση υπολογιστικών μονάδων με αρκετό χώρο και υπολογιστική δύναμη ώστε να χειριστούν τα δεδομένα, αλλιώς μπορούν να επιβραδυνθεί ή ακόμα και να τερματιστεί η εφαρμογή.

Ένα άλλο μειονέκτημα είναι, ορισμένες φορές η απαίτηση συγκεκριμένων μέσων παραγωγής πολυμέσων τα οποία δεν είναι ευρέως διαδεδομένα ή δεν είναι δημοσίως διαθέσιμα δωρεάν. Για παράδειγμα, η αναπαραγωγή ενός βίντεο .flv δεν μπορεί να αναπαραχθεί από οποιοδήποτε μέσο, αλλά ο χρήστης θα πρέπει να έχει στον υπολογιστή του εγκατεστημένο κάποιο μέσο που υποστηρίζει flash.

2.4 Παιχνίδια και Εκπαίδευση

Η διαδικασία εκμάθησης και εκπαίδευσης, καθότι πολύπλοκη, μπορεί αρκετές φορές να απωθήσει τους εκπαιδευόμενους λόγο έλλειψης ενδιαφέροντος και κινήτρου, οδηγώντας σε αντίθετα από τα επιθυμητά αποτελέσματα. Μέσα σε αυτό το πλαίσιο ο τομέας της εκπαίδευσης χρειάζεται ενίσχυση με μέσα τα οποία μπορούν να προσφέρουν κίνητρο στον εκπαιδευόμενο και παράλληλα να του κινούν το ενδιαφέρον. Οι υπολογιστές θεωρούνται ως το καταλληλότερο μέσο για την εμπλούτιση της διαδικασίας της εκπαίδευσης με υλικό ικανό να μαγνητίσει το ενδιαφέρον του εκπαιδευόμενου. Μία νεότερη τάση στον τομέα της εκπαίδευσης και του εκπαιδευτικού λογισμικού είναι τα επονομαζόμενα λογισμικά να συμπεριλαμβάνουν παιχνίδια τα οποία κινούνται σε εκπαιδευτικά πλαίσια. Μεταφέροντας την τάση αυτή στην ανθρώπινη καθημερινή ζωή, μπορούμε εύκολα να παρατηρήσουμε ότι τα παιχνίδια είναι εύκολο να προσελκύσουν το ενδιαφέρον διαφόρων ηλικιών ανθρώπων. Η χρήση των παιχνιδιών για εκπαιδευτικούς λόγους μπορεί να μετατρέψει τη διαδικασία της εκπαίδευσης σε ένα διασκεδαστικό τρόπο κατά τον οποίο δίνεται κίνητρο στον παίχτη να μάθει μέσω ενός διασκεδαστικού τρόπου.

Τα παιχνίδια συχνά επαινώντας για την δυνατότητα εκπαίδευσης που προσφέρουν. Αρκετές έρευνες έχουν αποδείξει ότι τα παιχνίδια στους υπολογιστές μπορούν να προσφέρουν μία νοητική άσκηση για τους παίχτες. Μέσω της δομής που τα παιχνίδια είναι σχεδιασμένα, μπορούν να αναπτύξουν μία σειρά νοητικών δυνατοτήτων στους παίχτες. Οι παίχτες αντιμετωπίζουν την ανάγκη ανάπτυξης μιας βραχυπρόθεσμης ή και μακροπρόθεσμης (εξαρτάται από το σκοπό του παιχνιδιού) σειράς αποφάσεων, και πρέπει να σχεδιάσουν στρατηγικές για επίλυση προβλημάτων (Johnson, 2005). Ο Papert(1993) σημειώνει ότι τα παιχνίδια μπορούν να προσφέρουν στρατηγικές εκπαίδευσης οι οποίες δίχως την ύπαρξη

παιχνιδιών είναι αδύνατο να εφαρμοστούν. Κάποια χαρακτηριστικά τέτοιων στρατηγικών είναι η ταχύρυθμη μετάδοση γνώσεων και η ανταμοιβή.

Ο Gee περιγράφει έναν κύκλο τεσσάρων βημάτων κατά τα οποία οι παίκτες εξετάζουν, αναπτύσσουν υποθέσεις, ξανά εξετάζουν και ξανά σκέπτονται. Μέσω αυτού του κύκλου αντικατοπτρίζονται στον παίχτη τα αποτελέσματα και οι επιπτώσεις που μπορούν να αποφέρουν οι αποφάσεις του και οι κινήσεις του κατά τη διάρκεια του παιχνιδιού. Ένας παρεμφερής κύκλος περιγράφεται από τους (Garris, Ahlers και Driskell, 2002) για παιχνίδια με εκπαιδευτικό περιεχόμενο. Κατά τη διάρκεια τέτοιων παιχνιδιών ο εκπαιδευόμενος επιδίδεται σε μία συνεχή επανάληψη κρίσης-γνώσεων-αποτελέσματος. Επιπρόσθετα οι (McFarlane, Sparrowhawk, και Heald, 2002) συνέδεσαν τα παιχνίδια με τη δυνατότητα ανάπτυξης δεξιοτήτων, αποφασιστικότητας, στρατηγικών, συνεργασίας και ικανότητας επίλυσης προβλημάτων. Στη εκδήλωση “Enhancing Learning; virtual worlds, simulations and games based learning” που πραγματοποιήθηκε από την BECTA, αναφέρθηκε ότι “Το ερώτημα δεν είναι πλέον εάν τα παιχνίδια μπορούν να βοηθήσουν στην εκπαίδευση, αλλά τι είδος εκπαίδευσης μπορούν να στηρίξουν και τι είδος εκμάθησης”. (BECTA, 2007).

Σαν αποτέλεσμα, αρκετά παιχνίδια επικεντρωμένα στην εκπαίδευση έχουν δημιουργηθεί, για παράδειγμα (Conati και Zhou 2002), (Virvou, Katsionis, και Manos, 2005). Τα αποτελέσματα αυτών των ερευνών, τα οποία κυρίως στηρίζονται στις εκπαιδευτικές δυνατότητες μαθητών σε σύγκριση με παραδοσιακές εκπαιδευτικές στρατηγικές, υπέδειξαν ότι οι μαθητές μπορούν να επωφεληθούν από εκπαιδευτικά παιχνίδια. Επιπρόσθετα από την αξιολόγηση αυτών των παιχνιδιών το συμπέρασμα είναι ότι οι μαθητές θα προτιμούσαν αυτόν τον τρόπο

εκπαίδευσης σε σύγκριση με τον παραδοσιακό καθώς χαρακτήρισαν τα παιχνίδια ως έναν ενδιαφέρον και διασκεδαστικό τρόπο μάθησης.

2.5 Τηλεπικοινωνίες και Εκπαίδευση.

Οι τεχνολογίες των τηλεπικοινωνιών έχουν ωφελήσει τον τομέα της εκπαίδευσης και στο χώρο των σχολείων αλλά και στις διάφορες εκπαιδευτικές διαδικασίες. Η χρήση τηλεπικοινωνιών στην εκπαίδευση έχει κεντρίσει το ενδιαφέρον, όχι μόνο για την εκπαίδευση των μαθητών στα σχολεία, αλλά και για την εξ'αποστάσεως εκπαίδευση διαφόρων ηλικιών εκπαιδευόμενων.

Ένας πολύ σημαντικός τομέας στον οποίο η τρέχουσα τεχνολογία κινητών τηλεπικοινωνιών μπορεί να ωφελήσει είναι η Εκπαίδευση. Στους γρήγορους ρυθμούς της καθημερινότητας, οι μαθητές αλλά και οι εκπαιδευτές θα εκτιμούσαν τη δυνατότητα του να μπορούν να συμμετέχουν ή αναπτύξουν (εξαρτάται το σκοπό) σε εκπαιδευτικά προγράμματα χωρίς να έχουν περιορισμούς στη γεωγραφική τους θέση. Δηλαδή, δυνατότητα εκπαίδευσης πέραν των συνόρων που θέτει ένας εκπαιδευτικό χώρος ή ένας σταθερός υπολογιστής.

Με βάση την αναγκαιότητα παροχής κινητών μεθόδων για εκπαιδευτικούς σκοπούς, αρκετοί ερευνητές έχουν στραφεί στην αναζήτηση μεθόδων που θα μπορέσουν να ικανοποιήσουν τις χαρακτηριστικά της κινητικότητας και της ανάγκης για μεταφορά. Πιο συγκεκριμένα, ανάμεσα στη βιβλιογραφία αναγνωρίζουμε ένα κινητό σύστημα εκμάθησης, το Mobile Author (Virvou και Alerpis, 2005) το οποίο έχει εφαρμογή και σε υπολογιστές, σε ταμπλέτες αλλά και σε κινητά τηλέφωνα.

2.6 Εξ' Αποστάσεως Εκπαίδευση --Distance learning

Η εκπαίδευση κατά την οποία οι εκπαιδευόμενοι και οι εκπαιδευτές δεν βρίσκονται στον ίδιο φυσικό χώρο, παραδοσιακά ονομάζεται εξ'αποστάσεως εκπαίδευση. Προδιαγραφές της εξ'αποστάσεως εκπαίδευσης είναι η ύπαρξη μέσων ικανών να μεταφέρουν στον εκπαιδευόμενο τη διδασκαλία γνώση. Καθώς οι τεχνολογίες συνεχώς βελτιώνονται και όλο και περισσότερες επικεντρώνονται στον τομέα της εκπαίδευσης, η εξ'αποστάσεως εκπαίδευση έχει αρχίσει και θεωρείται ως ένα από τους βασικούς τρόπους εκπαίδευσης.

Οι ηλεκτρονικοί υπολογιστές καθώς και ο παγκόσμιος ιστός, είναι από τα βασικά μέσα διαμόρφωσης της εξ'αποστάσεως εκπαίδευσης. Στην πραγματικότητα, η εκπαιδευτική αυτή μεθοδολογία λαμβάνει χώρα περισσότερο από 100 χρόνια, αναγνωρίζοντας στην Ευρώπη την πρώτη της εφαρμογή –υπό διαφορετική μορφή σε σύγκριση με τη σημερινή (Sherry, 1996). Η ανάπτυξη της εξ'αποστάσεως εκπαίδευσης έχει επισημανθεί από τη υιοθέτηση όλο και πιο εξελιγμένων τεχνολογιών και επικοινωνιών οι οποίες είναι διαθέσιμες στο ευρέως κοινό (Schlosser and Anderson 1994), ωθούμενη από το διαδίκτυο και τις τηλεδιασκέψεις, η εξ'αποστάσεως εκπαίδευση αποτελεί πλέον ένα ισχυρό εκπαιδευτικό μέσο. Weinstein 1997). Παρόλα αυτά, η εξ'αποστάσεως εκπαίδευση δεν βασίζεται κατά κόρον σε εφαρμογές βασισμένες στο διαδίκτυο, αλλά και σε εκπαιδευτικά λογισμικά ικανά να λειτουργήσουν χωρίς τη χρήση του (offline εφαρμογές).

Σε μία έρευνα για την εξ'αποστάσεως εκπαίδευση, (Schlosser και Anderson, 1994) από δεικνύεται ότι ένα από τους απώτερους σκοπούς είναι η δημιουργία μίας προσομοίωσης της

κλασικής εκπαίδευσης που λαμβάνει μέσα σε τάξη. Επιπρόσθετα, τονίζεται ότι η εξ'αποστάσεως εκπαίδευση δεν πρέπει να διαφέρει από τους παραδοσιακούς τρόπους εκμάθησης. Αφετέρου ο (Bates, 1995) υποστηρίζει ότι αντί να χρησιμοποιούμε τα τεχνολογικά επιτεύγματα για να προσομοιώσουμε υπάρχουσες διδασκαλικές μεθόδους, η χρήση της τεχνολογίας θα πρέπει να επικεντρώνεται στη βελτίωση και ανύψωση των μεθόδων αυτών.

2.7.1 Χαρακτηριστικά Εκπαιδευτικών Λογισμικών

Ως εκπαιδευτικό λογισμικό ορίζουμε οποιοδήποτε λογισμικό ή εφαρμογή έχει αναπτυχθεί με σκοπό η χρήση του να πραγματοποιείται μέσω κάποιου τεχνολογικού μέσου, με απώτερο σκοπό την εκμάθηση και την εκπαίδευση των χρηστών του σε κάποιο συγκεκριμένο γνωστικό πεδίο. Σύμφωνα με το (ITY, 2002) ως εκπαιδευτικό λογισμικό ορίζεται ένα εξειδικευμένο λογισμικό που μπορεί να υποστηρίζει μαθησιακές και διδακτικές δραστηριότητες μέσα στα πλαίσια συγκεκριμένων προγραμμάτων σπουδών. Απώτερος σκοπός είναι η μέγιστη δυνατή αξιοποίηση των υπάρχουσών τεχνολογιών για τη δημιουργία ενός ελκυστικού για τους εκπαιδευόμενους μαθησιακού περιβάλλοντος (Παπαδόπουλος, 2001).

Προκειμένου να κατηγοριοποιηθούν τα διάφορα είδη εκπαιδευτικού λογισμικού, έχουν προταθεί διάφορες ομάδες κριτηρίων. Σύνηθες κριτήρια είναι ο βαθμός χρήσης του λογισμικού στη διδακτική διαδικασία, ο βαθμός αλληλεπίδρασης με τον εκπαιδευόμενο, οι διδακτικές προσεγγίσεις που ακολουθεί το διδακτικό. Στη βιβλιογραφία βρίσκουμε τις

ακόλουθες κατηγορίες:

- ▲ Κατηγοριοποίηση με βάση τις υποκείμενες θεωρίες μάθησης και τις συνεπαγόμενες διδακτικές πρακτικές (EAITY, 2007α). Σε αυτήν την κατηγορία ανήκουν λογισμικά που ακολουθούν πρακτικές καθοδηγούμενης διδασκαλίας (tutoring systems) καθώς και λογισμικά τα οποία παρέχουν στον χρήστη τη δυνατότητα επικοινωνίας, έκφρασης και δημιουργίας. Το σύνηθες είναι, λογισμικά που ανήκουν σε αυτή την κατηγορία, να ακολουθούν συμπεριφορές , γνωστικές και κοινωνικοπολιτιστικές της θεωρίας μάθησης.
- ▲ Κατηγοριοποίηση με βάση τις τεχνολογίες ανάπτυξης και τα παιδαγωγικά ρεύματα. Σε αυτήν την κατηγορία το λογισμικό αναλαμβάνει κάποιον ρόλο. Το λογισμικό μπορεί να πράξει ως ο εκπαιδευτής του χρήστη με αποτέλεσμα ο ρόλος του να είναι καθοδηγητικός, μπορεί να πράξει σαν εκπαιδευόμενος με αποτέλεσμα ο ρόλος του να είναι συμβατικός, είτε να πράξει σαν συνεργάτης του εκπαιδευόμενου έτσι ώστε ο ρόλος του να είναι συμβουλευτικός.
- ▲ Κατηγοριοποίηση με βάση τη χρήση του λογισμικού στη μαθησιακή διαδικασία (ΠΙ, 2007). Κατά την ανάπτυξη του λογισμικού διαφορετικοί παράμετροι λαμβάνονται υπόψιν, καθώς στοχεύουν σε διαφορετικές χρήσεις. Σε πιο λεπτομέρεια, στη βιβλιογραφία αναφέρονται τα λογισμικά τα οποία επικεντρώνονται σε εφαρμογές για την εξάσκηση του χρήστη, σε λογισμικά τα οποία προορίζονται για παρουσιάσεις διαλέξεων, σε λογισμικά τα οποία έχουν ως στόχο την προσομοίωση διδακτικών διαδικασιών. Οι προσομοιώσεις στην εκπαίδευση μπορούν να αναφέρονται είτε στην ερευνητική διαδικασία είτε στην διδακτική διαδικασία αν και συχνά αυτές οι διαδικασίες επικαλύπτονται και τροφοδοτούν η μία την άλλη. (Ψυχάρης & Φραγκάκης , 2006). Επίσης διακρίνουμε και τα λογισμικά τα οποία επικεντρώνονται

σε τεχνικές επίλυσης προβλημάτων και αξιολόγησης, και τέλος στα παιχνίδια με διδακτικούς σκοπούς.

- ▲ Κατηγοριοποίηση με βάση το βαθμό αλληλεπίδρασης με το χρήστη. Σε αυτήν την κατηγορία αναγνωρίζουμε τα ανοιχτά εκπαιδευτικά περιβάλλοντα τα οποία δίνουν απόλυτη ελευθερία στο χρήστη, καθώς και τα κλειστά περιβάλλοντα κατά τα οποία η ελευθερία του χρήστη είναι περιορισμένη.
- ▲ Κατηγοριοποίηση με βάση τον εκπαιδευτικό στόχο του λογισμικού. Ένα εκπαιδευτικό λογισμικό μπορεί να εξυπηρετεί σκοπούς για την ανάπτυξη της γραπτής έκφρασης του χρήστη (κειμενογράφος), ή την παρουσίαση γνωστικής ύλης με χρήση πολυμέσων (εικόνα, ήχος, βίντεο).
- ▲ Κατηγοριοποίηση με βάση τη παιδαγωγική προσέγγιση. Το λογισμικό μπορεί να εισάγει και να υποστηρίζει νέους τρόπους διδασκαλίας. Με αυτόν τον τρόπο βοηθά τους εκπαιδευόμενους να εξασκήσουν υπάρχουσες δεξιότητες αλλά και να αναπτύξουν νέες. Επίσης αυτή η μέθοδος έχει ως σκοπό την όξυνση της αντίληψης και της κριτικής σκέψης του εκπαιδευόμενου. Μία άλλη κατηγορία λογισμικού είναι αυτή στην οποία συναντάμε τις κλασικές μεθόδους διδασκαλίας οι οποίες έχουν ως σκοπό την εκγύμναση και την πρακτική εξάσκηση του εκπαιδευόμενου.

2.7.2 Αλληλεπιδραστικά Εκπαιδευτικά Περιβάλλοντα

Ως αλληλεπιδραστικά εκπαιδευτικά περιβάλλοντα αναγνωρίζουμε τα ανοιχτά περιβάλλοντα που επιτρέπουν στο χρήστη να ενεργήσει ελεύθερα, αξιοποιώντας τις δυνατότητες του

λογισμικού, χωρίς να διαθέτουν κάποια διαδικασία που να τον κατευθύνει. Ο χρήστης είναι ελεύθερος να επιλέξει τον τρόπο και το χρόνο πλοήγησής του στο λογισμικό, την ύλη που θα διδαχθεί. Επιπρόσθετα σε αυτήν την κατηγορία περιβαλλόντων δημιουργείται στο χρήστη η αίσθηση επικοινωνίας με το λογισμικό, δίνοντας μία πιο “φιλική” προσέγγιση. Η αναπαράσταση της γνώσης, συνήθως στα αλληλεπιδραστικά περιβάλλοντα, είναι πολυδιάστατη υπό την έννοια ότι μπορεί να πάρει ποικίλλες μορφές δίνοντας τη δυνατότητα στο χρήστη να επιλέξει αυτή που του αρμόζει καλύτερα.

Το είδος αυτό εκπαιδευτικών περιβαλλόντων είναι επικεντρωμένο στο χρήστη και στις ανάγκες του χρήστη, έτσι ώστε να δημιουργήσει ένα εξατομικευμένο σύστημα εκμάθησης ικανό να κεντρίσει το ενδιαφέρον του χρήστη.

2.7.3 Αξιολόγηση Εκπαιδευτικού Λογισμικού

Ένα εκπαιδευτικό λογισμικό μπορεί να αξιολογηθεί ως προς το περιεχόμενό του, την διδακτική του προσέγγιση καθώς και τις τεχνολογίες που χρησιμοποιεί. Πιο λεπτομερές, η αξιολόγηση με βάση το περιεχόμενο και τη διδακτική προσέγγιση γίνεται με τα έξι κριτήρια:

- ▲ Σε τι βαθμό καλύπτεται η επιθυμητή διδακτέα ύλη.
- ▲ Κατά πόσο το περιεχόμενο της διδαχθείσας ύλης έχει αξιοπρεπής παρουσίαση, υπό την έννοια χρήσης σωστών ορισμών και σωστής γραμματικά και συντακτικά γλώσσας.
- ▲ Αν η ποσότητα της διδαχθείσας ύλης αντικατοπτρίζεται από τον διαθέσιμο χρόνο του

εκπαιδευόμενου.

- ⤴ Κατά πόσο η διδακτέα ύλη ανταποκρίνεται στο μαθησιακό επίπεδο του εκπαιδευόμενου.
- ⤴ Κατά πόσο η διδακτική προσέγγιση προκαλεί το ενδιαφέρον του εκπαιδευόμενου έτσι ώστε να είναι αποτελεσματική και ελκυστική.

Η αξιολόγηση της τεχνολογίας του εκπαιδευτικού λογισμικού γίνεται με βάση τα εξής κριτήρια:

- ⤴ Το βαθμό αξιοποίησης των δυνατοτήτων για αλληλεπίδραση.
- ⤴ Το βαθμό ελέγχου που προσφέρει το λογισμικό για τον έλεγχο της εκπαιδευτικής διαδικασίας.
- ⤴ Κατά πόσο το περιεχόμενο είναι εμπλουτισμένο υπό την έννοια χρήσεως πολυμέσων (εικόνα, ήχος, βίντεο, κ.τ.λ.).

Αρκετές μελέτες και δημοσιεύσεις αναφέρονται στα χαρακτηριστικά και τις προδιαγραφές που τα εκπαιδευτικά λογισμικά πρέπει να πληρούν. Ο σχεδιασμός του εκπαιδευτικού λογισμικού πρέπει να απαντάει στις εξής τρεις ερωτήσεις (Laurillard 1993):

- ⤴ Πώς μεταφέρεται η γνώση μέσω του λογισμικού;
- ⤴ Ποιο είναι το εκπαιδευτικό αντικείμενο και γιατί διδάσκεται;
- ⤴ Σε ποιόν απευθύνεται;

Ακολουθώντας αυτά τα τρία στάδια σχεδίασης του λογισμικού, και σύμφωνα με τις νεότερες ψυχολογικές και παιδαγωγικές θεωρίες εκμάθησης, το εκπαιδευτικό λογισμικό πρέπει να στοχεύει σε συγκεκριμένο ακροατήριο και να λαμβάνει υπ' όψιν

τα γνωστικά επίπεδα των χρηστών του –κοινώς το υπόβαθρο του χρήστη σε σχέση με το μαθησιακό αντικείμενο του λογισμικού.

Προκειμένου ένα εκπαιδευτικό λογισμικό να θεωρείται ενδιαφέρον και ελκυστικό για τους χρήστες του, σύμφωνα με την έρευνα που παρουσιάζεται στο (Παπαδόπουλος, Πολάτογλου 2002) πρέπει να ακολουθεί τα παρακάτω χαρακτηριστικά:

- ♣ Διεπιστημονικότητα.
- ♣ Διευρηνητικότητα.
- ♣ Πολλαπλή αναπαράσταση της ίδιας έννοιας.
- ♣ Ολοκληρωμένη παρουσίαση αρχών και νόμων.
- ♣ Αυτοαξιολόγηση.
- ♣ Φιλικότητα.
- ♣ Φωνή/ήχο.
- ♣ Να εντυπωσιάζει.
- ♣ Αλληλεπίδραση.
- ♣ Φιλικό στο χρήστη.

2.8 Συμπέρασμα

Σε αυτήν την ενότητα παρουσιάσαμε συνοπτικά θεωρήσεις, έρευνες και αποτελέσματα ερευνών στον τομέα της εκπαίδευσης και πιο συγκεκριμένα της ηλεκτρονικής εκπαίδευσης. Οι αρχές που ακολουθήσαμε για το σχεδιασμό και την ανάπτυξη του εκπαιδευτικού λογισμικού το οποίο παρουσιάζουμε, είναι εξολοκλήρου βασισμένο στις αρχές που αναφέρουμε στο κεφάλαιο αυτό.

Πιο συγκεκριμένα, παρουσιάζουμε την ανάπτυξη ενός διαδραστικού εκπαιδευτικού λογισμικού, του οποίου η χρήση μπορεί να εξυπηρετήσει online αλλά και offline σκοπούς. Η διαδικασία εκμάθησης που ακολουθούμε είναι επικεντρωμένη στο χρήστη, υπό την έννοια ότι έχει σχεδιαστεί για να καλύπτει τις ανάγκες του και τυχόν μαθησιακές δυσκολίες, αλλά και να του παρέχει την απαραίτητη ελευθερία κατά τη διάρκεια της χρήσης του λογισμικού. Παράλληλα, κάνουμε χρήση πολυμέσων με σκοπό του εμπλουτισμού του εκπαιδευτικού υλικού. Πέραν των καθιερωμένων μεθόδων, κείμενο-εικόνα, κάνουμε χρήση ηχογραφημένων διαλέξεων, μουσικής, βίντεο αλλά και διαδραστικών εφαρμογών όπως είναι τα παιχνίδια, τα quiz και τα τεστ, προκειμένου να δημιουργήσουμε ένα πιο ενδιαφέρον σύστημα εκμάθησης.

Κεφάλαιο 3. Επιλέγοντας την Κατάλληλη Τεχνολογία

3.1 Εισαγωγή

Η σημερινή κοινωνία της πληροφορίας, καθιστά αναγκαίο από τους εκπαιδευόμενους να είναι συμφιλωμένοι με κάποιες τεχνολογικές αρχές και επιτεύγματα. Μία μεγάλη αφθονία από τεχνολογίες για κατασκευή λογισμικού, είναι διαθέσιμη στους προγραμματιστές, οι οποίοι με βάση τις ανάγκες του λογισμικού προς δημιουργία, μπορούν να επιλέξουν τις καταλληλότερες. Στη δουλειά του Kitao (1997), βρίσκουμε τον ορισμό των «υλικών», δίνοντας έμφαση στη δυνατότητα αυτών των υλικών να ελέγξουν και να βοηθήσουν στη διαδικασία της εκμάθησης και της διδασκαλίας. Τα αναφερόμενα υλικά αντιστοιχούν σε κείμενα, βίντεο, ηχογραφήσεις, λογισμικό σε υπολογιστές και σε οπτικό υλικό. Η επιλογή του υλικού αντανακλά στην αναπαράσταση της γνώσης προς εκμάθηση.

Ένα υψηλής ποιότητας εκπαιδευτικό λογισμικό μπορεί να έχει πολλούς ορισμούς. Ορισμένοι θεωρούν τα προγράμματα που είναι απαλλαγμένα από προγραμματιστικά λάθη ως επαρκής ποιότητας λογισμικά για τους εκπαιδευόμενους. Άλλοι υποστηρίζουν ότι ένα εκπαιδευτικό λογισμικό μπορεί να χαρακτηριστεί σαν υψηλής ποιότητας λογισμικό αρκεί να πληρεί τις εκμαθησιακές και διδασκαλικές προδιαγραφές. Μπορεί να θεωρηθεί ως υψηλής ποιότητας ένα λογισμικό το οποίο εκτελεί διδασκαλικούς σκοπούς αλλά είναι δύσκολο στη χρήση ; Παίζει ρόλο η φιλικότητα προς το χρήστη; Για τη δουλειά που παρουσιάζουμε σε αυτή τη διατριβή, έχουμε θεωρήσει ότι ένα υψηλής ποιότητας εκπαιδευτικό λογισμικό πρέπει να πληρεί τις παρακάτω προδιαγραφές:

- Να είναι απαλλαγμένο από προγραμματιστικά λάθη.
- Να είναι απαλλαγμένο από λογικά λάθη.
- Να ακολουθεί τις εκπαιδευτικές αρχές που αναφέραμε στο προηγούμενο κεφάλαιο.
- Να είναι ελκυστικό για τον εκπαιδευόμενο.
- Να είναι φιλικό προς το χρήστη.

Σε πιο λεπτομέρεια, όσο αναφορά τη φιλικότητα προς το χρήστη και το πόσο ελκυστικό μπορεί να είναι , βασιζόμαστε στις προδιαγραφές που αναγνωρίζονται στη δουλειά των (Παπαδόπουλος, Πολάτογλου 2002). Οι προδιαγραφές είναι:

- Πολλαπλή αναπαράσταση της ίδιας έννοιας
- Ολοκληρωμένη παρουσίαση αρχών και νόμων
- Αυτοαξιολόγηση
- Φιλικότητα
- Φωνή/ήχο
- Να εντυπωσιάζει
- Αλληλεπίδραση
- Φιλικό στο χρήστη

Επομένως οι τεχνολογίες που θα επιλεγθούν για την ανάπτυξη του εκπαιδευτικού λογισμικού, πρέπει να προσφέρουν στον προγραμματιστή την ευκολία ανάπτυξης εφαρμογών οι οποίες να αντιστοιχούν στις παραπάνω προδιαγραφές. Χρειαζόμαστε μια γλώσσα ή ένα προγραμματιστικό περιβάλλον τέτοιο ώστε:

- Να μπορούμε να αναπτύξουμε και να διαχειριστούμε πολυμεσικό υλικό
- Να δημιουργήσουμε την ψευδαίσθηση της τάξης
- Να δημιουργήσουμε την ψευδαίσθηση του βιβλίου
- Να είναι ευρέως διαδεδομένο
- Να επιτρέπει τη δημιουργία εφαρμογών εύκολων στη χρήση
- Να επιτρέπει τη δημιουργία διαδραστικών εφαρμογών
- Να υποστηρίζει on-line και off-line εφαρμογές
- Να προσφέρει τη δυνατότητα εύκολου εντοπισμού προγραμματιστικών και λογικών λαθών (debugging)

Περαιτέρω, το εκπαιδευτικό λογισμικό που θα αναπτυχθεί θα πρέπει να παρέχει τη δυνατότητα συντήρησης και πιθανό τροποποιήσεις στο μέλλον. Γεγονός το οποία καθιστά και τρεις περεταίρω προδιαγραφές :

- Η προγραμματιστική γλώσσα ή το προγραμματιστικό περιβάλλον που θα χρησιμοποιηθεί να είναι διαδεδομένη. Εάν το λογισμικό αναπτυχθεί σε μία άκρως εξιδανικευμένη γλώσσα ή προγραμματιστικό περιβάλλον, και ο προγραμματιστής που το δημιούργησε δεν λάβει μέρος σε μελλοντικές τροποποιήσεις ή συντήρηση του λογισμικού, τότε καθίσταται απαραίτητο να βρεθεί κάποιος άλλος προγραμματιστής

με παρεμφερείς γνώσεις στο συγκεκριμένο αντικείμενο –και εφόσον αναφερόμαστε σε εξειδικευμένες γνώσεις, η εύρεση κάποιου άλλου προγραμματιστή θα είναι χρονικά χρονοβόρα.

- Το τελικό λογισμικό να βασίζεται σε κάποια ευρέως διαδεδομένη τεχνολογία, η οποία να μπορεί να αναπαραχθεί από τα περισσότερα μέσα. Υπό αυτήν την προϋπόθεση, το λογισμικό θα πρέπει να μπορεί να αναπαραχθεί σε ένα μηχάνημα ενός μέσου χρήστη, και να μην απαιτεί την εγκατάσταση περεταίρω λογισμικού ή περεταίρω υπολογιστικούς πόρους.
- Η ανάπτυξη του λογισμικού να γίνει σε μία τρέχουσα τεχνολογία, προς αποφυγήν δημιουργίας ενός legacy system όσο αναφορά μελλοντικές τροποποιήσεις.

3.2.1 Τεχνολογία Flash

Η τεχνολογία flash χρησιμοποιεί vector και raster γραφικά για τη δημιουργία κίνησης σε κείμενα και εικόνες. Το flash υποστηρίζει αμφίδρομη ροή ήχου και εικόνας και μπορεί να προσφέρει διαδραστικότητα με το χρήστη παίρνοντας σαν input δεδομένα από το ποντίκι, το πληκτρολόγιο, το μικρόφωνο ή ακόμα και την κάμερα του χρήστη. Το flash περιλαμβάνει μία αντικειμενοστραφής γλώσσα προγραμματισμού, Actionscript, και υποστηρίζει την αυτοματοποίησή της μέσω του JavaScript Flash language (JSFL).

Το περιεχόμενο του flash μπορεί να αναπαραχθεί από πολλά μέσα, όπως οι υπολογιστές, ένα μεγάλο μέρος κινητών και ένα μεγάλο μέρος ταμπλέτων, με τη χρήση κάποιου player, όπως

ο Adobe Flash Player ¹. Το λογισμικό το οποίο αναπτύσσεται σε τεχνολογία Flash, συνήθως είναι εξολοκλήρου βασισμένο σε αυτή την τεχνολογία.

Η τεχνολογία Flash παρέχει τις δυνατότητες:

- Δημιουργίας κινούμενων οπτικών εφέ.
- Δημιουργίας αξιόπιστης εμφάνισης στα περισσότερα μέσα αναπαραγωγής.
- Δημιουργίας εντυπωσιακού περιεχομένου, υπό την έννοια των γραφικών.
- Ευελιξίας και ταχύτητας φόρτωσης.

3.2.2 Χαρακτηριστικά του Flash

Σε αντίθεση με τις αρχικές ημέρες του διαδικτύου, όταν οι ιστοσελίδες βασίζονταν κυρίως σε κείμενο ή κείμενο με μερικές στατικές εικόνες, οι ιστοσελίδες και γενικότερα οι εφαρμογές, on-line και off-line, σήμερα είναι δυναμικές και διαθέτουν μία ποικιλία μέσων από εικόνες, βίντεο και διαδραστικών εφαρμογών. Το Flash επιτρέπει την ανάπτυξη διαδραστικών και πολυμεσικών εφαρμογών, έτσι ώστε οι προγραμματιστές να μπορούν να έχουν πρόσβαση σε περισσότερους πελάτες, χωρίς την ανάγκη για προγραμματισμό και συνδυασμό περισσότερων από μίας γλώσσας προγραμματισμού.

Το Flash είναι μία από τις κορυφαίες τεχνολογίες σχεδιασμού σήμερα διαθέσιμες στους περισσότερους κατασκευαστές. Είναι μια πλατφόρμα που δίνει στους προγραμματιστές περισσότερες επιλογές σε σχέση με άλλες πλατφόρμες. Δίνει τη δυνατότητα δημιουργίας οπτικών εφέ αιχμής τα οποία να υποστηρίζουν πλήρως κινούμενη πλοήγηση, έτσι ώστε να

¹ <http://www.adobe.com/products/flashplayer.html>

δώσει στον τελικό χρήστη μια πραγματικά ζωντανή και διαδραστική εμπειρία. Επιπλέον τα πολυμεσικά περιεχόμενα, ήχος – εικόνα – βίντεο συνδυάζονται ομαλά με κίνηση, καθώς μπορούν να υποστηριχθούν με επιτυχία και να φορτώσουν σε ελάχιστο χρόνο –γεγονός που καθιστά εφικτή τη δημιουργία υψηλών προδιαγραφών εφαρμογών.

3.2.3 Προτερήματα του Flash

Ελευθερία

Το Flash παρέχει ένα μεγάλο βαθμό ελευθερίας κατά τη διάρκεια ανάπτυξης εφαρμογών/λογισμικού, στους προγραμματιστές, υπό την έννοια ότι επιτρέπει την τοποθέτηση αντικειμένων ² σε οποιοδήποτε σημείο θέλει ο προγραμματιστής, χωρίς να ανησυχεί για ασυνέπειες και λογικά λάθη. Ο προγραμματιστής έχει έναν «καμβά», πάνω στον οποίο τοποθετεί οπουδήποτε θέλει τα αντικείμενα του.

Διαδραστικότητα

Παρέχει ένα διαδραστικό περιβάλλον για μια πιο περίπλοκη εμπειρία πολυμέσων. Δεν αφορά απλά δυνατότητα αλλαγής χρωμάτων ή εναλλαγής εικόνων, αλλά επιτρέπει το συνδυασμό ήχου και βίντεο.

Έλεγχος προσαρμογής

Η απόδοση των διανυσματικών εικόνων του Flash δίνει τη δυνατότητα προσαρμογής ολόκληρης της εικόνας προς επίδειξη με βάση το μέγεθος της οθόνης στην οποία προβάλλεται. Αυτό είναι ένα ιδιαίτερο προτέρημα για προβολή εφαρμογών σε μικρές

² Υποσημείωση ότι το Flash βασίζεται στη αντικειμενοστραφής προγραμματιστική προσέγγιση.

συσκευές, όπως τα κινητά τηλέφωνα ή η ταμπλέτες, κατά τα οποία τις περισσότερες φορές επιθυμούμαι να μεγεθύνουμε την εικόνα/κείμενο κ.λπ. που προβάλλονται.

Συνδυασμός Πολυμέσων

Το Flash επιτρέπει το συνδυασμό οποιουδήποτε τύπου πολυμέσου, σε μία κοινή εφαρμογή. Για παράδειγμα, μπορούμε να έχουμε στην ίδια εφαρμογή bitmap εικόνες σε διάφορους τύπους (όπως για παράδειγμα, GIF, JPEG, PNG, PCT, TIF), διανυσματικές εικόνες (συμπεριλαμβανομένου FreeHand files, EPS, Illustrator files), και όπως αναφέραμε και προηγούμενος, διάφορους τύπους ήχου (WAV, AIF, MP3). Ο συνδυασμός όλων αυτών μπορεί να εισαχθεί σε μία Flash εφαρμογή.

Αναπαραγωγή Frames³

Σε περιβάλλον ανάπτυξης Flash μπορούμε να αναπαράγουμε τη συμπεριφορά οποιουδήποτε frame, χωρίς να χρειάζεται να επαναπρογραμματίσουμε ή να ξανασχεδιάσουμε κάποιο από τα στοιχεία και τα χαρακτηριστικά του frame.

Δεν χρειάζεται επαναφόρτωση

Το Flash δίνει τη δυνατότητα να προβάλλουμε δεδομένα σαν κομμάτι μίας εναλλασσόμενης διαδικασίας, έτσι ώστε να μην χρειάζεται να επαναφορτώσουμε την εφαρμογή ή να ανανεώσουμε την εφαρμογή. Η κάθε αλλαγή που κάνει ο προγραμματιστής στην εφαρμογή, εμφανίζεται δυναμικά.

³ Frame (καρέ) - Είναι η μικρότερη μονάδα μεταβολής ενός αντικειμένου ή ενός συμβόλου στο διάγραμμα ροής χρόνου μιας ταινίας. Στις συμβατικές κινηματογραφικές ταινίες, το καρέ είναι μία από τις αναρίθμητες στατικές φωτογραφίες, οι οποίες, όταν προβάλλονται διαδοχικά και με τη σωστή ταχύτητα, δίνουν την αίσθηση της κίνησης. Τα καρέ-κλειδιά (keyframes) είναι τα σημεία στα οποία μπορούμε να ορίσουμε εμείς οποιαδήποτε μεταβολή, όπως την αλλαγή του μεγέθους ή της πορείας ενός αντικειμένου και μετά να πούμε στο Flash να δημιουργήσει αυτόματα μόνο του τα ενδιάμεσα καρέ ανάμεσα στα καρέ-κλειδιά.

3.2.4 Αδυναμίες του Flash

Απαιτεί την εγκατάσταση λογισμικού

Οι Flash εφαρμογές απαιτούν την εγκατάσταση κάποιου παράταιρου λογισμικού, όπως κάποιου flash plug-in προκειμένου να αναπαραχθεί σε κάποιον browser, ή την εγκατάσταση κάποιου κατάλληλου player προκειμένου να αναπαραχθεί στον υπολογιστή. Από τη στιγμή όπου το απαραίτητο προς εγκατάσταση λογισμικό είναι ελεύθερο (δεν απαιτεί κάποια πληρωμή ή συνδρομή), η απαίτηση προς εγκατάστασή του δεν θεωρείται μεγάλο μειονέκτημα. Ωστόσο, λαμβάνοντας υπόψιν ότι σε αρκετές περιπτώσεις ο χρήστης δεν έχει την ελευθερία να εγκαταστήσει το λογισμικό της αρεσκείας του, για παράδειγμα σε εργασιακά περιβάλλοντα, ή σε σχολικά, σε αυτές τις περιπτώσεις οι Flash εφαρμογές μειονεκτούνε.

Αναζήτηση Περιεχομένου

Οι τυπικές μηχανές αναζήτησης στο διαδίκτυο αδυνατούν να εντοπίσουν το περιεχόμενο που είναι αποθηκευμένο σε Flash εφαρμογές. Αυτό έχει ως αποτέλεσμα, για ένα 100% βασισμένο σε Flash διαδικτυακό λογισμικό να αδυνατεί να προβάλλει το υλικό του μέσω των μηχανών αναζήτησης, το οποίο μπορεί να θεωρηθεί ένα από τα μεγαλύτερα μειονεκτήματα του Flash.

Λειτουργικά για κινητή τηλεφωνία, ταμπλέτες και υπολογιστές

Αν και το Flash υποστηρίζεται από ένα μεγάλο μερίδιο της αγοράς λογισμικού για κινητή τηλεφωνία και ταμπλέτες, στο υπόλοιπο ποσοστό δεν υποστηρίζεται. Ο λόγος γίνεται για το λογισμικό που παρέχεται από την εταιρεία Apple⁴, η οποία είναι υπεύθυνη για ένα ευκαταφρόνητο μερίδιο της αγοράς κινητής τηλεφωνία και ταμπλετών. Παρόλο που οι

⁴ <http://www.apple.com>

υπόλοιπες εταιρείες υποστηρίζουν την αναπαραγωγή Flash υλικού, σε μία πρόσφατη ανακοίνωση της Adobe⁵ υποστηρίζει ότι η επόμενη έκδοση του Flash Player, 11.2, που αναμένεται σε λίγες βδομάδες θα αποτελέσει την τελευταία έκδοση Flash που θα είναι διαθέσιμη για Linux. Μετά από αυτή, ο μόνος τρόπος υποστήριξης του Flash σε Linux περιβάλλον θα γίνεται μόνο μέσω του Pepper, ένα API που αναπτύχθηκε με συνεργασία των Adobe και Google. Το Pepper υποστηρίζεται μόνο από τον browser Google Chrome. Κοινώς, μετά την έκδοση 11.2 αν κάποιος χρήστης Linux θελήσει Flash θα πρέπει αναγκαστικά να χρησιμοποιήσει τον Google Chrome browser. Το αποτέλεσμα είναι, ότι οι Flash εφαρμογές θα είναι διαθέσιμες μόνο στο ποσοστό των χρηστών Linux που χρησιμοποιούν τον Google Chrome browser, είτε σε κινητά τηλέφωνα είτε σε υπολογιστές.

3.2.5 Το μέλλον του Flash

Στην επίσημη ανακοίνωση της Adobe⁶ ο πρόεδρος της εταιρείας Danny Winokur έγραψε ότι το HTML5⁷ είναι πλέον παγκόσμιος αποδεκτό από τις περισσότερες συσκευές -- σταθερές και κινητές, καθώς σε μερικές περιπτώσεις είναι αποκλειστική η χρήση του. Αυτό κάνει το HTML5 μία από τις καλύτερες λύσεις για ανάπτυξη διαδικτυακών εφαρμογών. Είμαστε ενθουσιασμένοι για αυτό, και θα συνεχίσουμε τη συνεισφορά μας στην HTML κοινότητα και σε εταιρείες-κλειδιά όπως η Google, Microsoft, Apple και RIM, έτσι ώστε να προσφέρουμε HTML5 καινοτομίες που θα τους βοηθήσουν να προωθήσουν τις τεχνολογίες τους".

Λαμβάνοντας υπόψιν ότι στην αγορά εμφανίζονται όλο και περισσότερες συσκευές και λογισμικά τα οποία δεν υποστηρίζουν Flash εφαρμογές, σε συνδυασμό με την άνοδο των

⁵ <http://www.adobe.com/devnet/flashplatform/whitepapers/roadmap.html>

⁶ <http://blogs.adobe.com/conversations/2011/11/flash-focus.html>

⁷ Hypertext Markup Language για ανάπτυξη δικτυακών και μη εφαρμογών με πλούσιο διαδραστικό περιεχόμενο.

HTML5 τεχνολογιών, πολλά άρθρα αναφέρονται στην αντικατάσταση του Flash από το HTML5, όπως (Lee,2010 και Vaughan-Nichols, 2010).

3.3. Συμπέρασμα

Σε αυτό το κεφάλαιο παρουσιάσαμε τα χαρακτηριστικά του εκπαιδευτικού λογισμικού προς ανάπτυξη, λαμβάνοντας υπόψιν τις υπάρχουσες τεχνολογίες. Ο σχεδιασμός και η ανάπτυξη του εκπαιδευτικού λογισμικού που αναπτύξαμε βασίστηκε εξολοκλήρου στη Flash τεχνολογία. Πιστεύουμε, ότι για τις σημερινές τεχνολογικές ανάγκες των λογισμικών τα οποία πρέπει να παρέχουν ένα πλούσιο και διαδραστικό υλικό στο χρήστη, το Flash παρέχει την πιο αξιόπιστη, σταθερή και διαδεδομένη λύση. Παρόλα αυτά, στηρίζουμε την ιδέα της ανάπτυξης εφαρμογών σε ανοιχτές τεχνολογίες όπως το HTML5, και συμεριζόμαστε την ανάγκη για μία ενιαία τεχνολογία την οποία να μπορούν, αν όχι όλες --οι περισσότερες συσκευές να στηρίζουν. Ωστόσο, πιστεύουμε ότι η αντικατάσταση του Flash από το HTML5 δεν θα πραγματοποιηθεί στο κοντινό μέλλον.

Κεφάλαιο 4. Προγραμματιστικές Γλώσσες

4.1. Εισαγωγή

Σε αυτό το κεφάλαιο θα αναφέρουμε κάποια από τα βασικά χαρακτηριστικά των γλωσσών προγραμματισμού που χρησιμοποιήσαμε κατά την ανάπτυξη του εκπαιδευτικού λογισμικού για Δίκτυα Υπολογιστών και Τηλεπικοινωνιών. Ο σκελετός του εκπαιδευτικού λογισμικού, καθώς και οι λειτουργίες του είναι εξολοκλήρου αναπτυγμένα σε Action Script 3. Παρόλα αυτά, έχει γίνει χρήση των τεχνολογιών HTML, CSS, XML για τη δημιουργία μίας ομορφότερης προβολής των δεδομένων.

4.2.1 Action Script

Η Action Script είναι μία ανοιχτού-λογισμικού γλώσσα που επικεντρώνεται σε κινούμενες εικόνες και σε διαδραστικό υλικό. Πρωταρχικός σκοπός της χρήσης ήταν η ανάπτυξη ιστοσελίδων και λογισμικού για τη πλατφόρμα του Adobe Flash Player.

Η Action Script είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού, η οποία αρχικά αναπτύχθηκε από τη Macromedia⁸. Είναι μία διάλεκτος της ECMAScript -- δηλαδή είναι ένα υπερσύνολο του συντακτικού, της σημασιολογίας και της γλώσσας που είναι ευρέως διαδεδομένη σαν Java Script: αποτελείται από κλάσεις (class), αντικείμενα (object), καθώς ο προγραμματιστής μπορεί να συμπεριλάβει (include) πακέτα (packages) τα οποία προσφέρουν έτοιμες κλάσεις και συναρτήσεις.

4.2.2 Κλάσεις και Αντικείμενα

Κεντρική ιδέα στον αντικειμενοστραφή προγραμματισμό είναι η κλάση (class). Η κλάση είναι μία αυτοτελής και αφαιρετική αναπαράσταση κάποιας κατηγορίας αντικειμένων, φυσικών είτε νοητών. Δηλαδή, προσδιορίζει τον τύπο των δεδομένων ή αλλιώς το προσχέδιο μίας δομής δεδομένων με τα δικά της περιεχόμενα (μεταβλητές και διαδικασίες). Τα περιεχόμενα αυτά μπορούν να δηλωθούν είτε ως δημόσια (public) τα οποία μπορούν να είναι προσπελάσιμα και από κώδικα εκτός κλάσης, είτε ως ιδιωτικά (private). τα οποία είναι προσπελάσιμα μόνο από τον κώδικα της κλάσης.

⁸ Η Macromedia πλέον είναι συγχωνευμένη με την Adobe

Οι διαδικασίες των κλάσεων συνήθως καλούνται μέθοδοι (methods) και οι μεταβλητές τους γνωρίσματα (attributes) ή πεδία (fields). Μία κλάση πρέπει ιδανικά να είναι εννοιολογικά αυτοτελής, να περιέχει δηλαδή μόνο πεδία τα οποία περιγράφουν μία κατηγορία αντικειμένων και δημόσιες μεθόδους οι οποίες επενεργούν σε αυτά όταν καλούνται από το εξωτερικό πρόγραμμα, χωρίς να εξαρτώνται από άλλα δεδομένα ή κώδικα εκτός της κλάσης, και επαναχρησιμοποιήσιμη.

Αντικείμενο (object) είναι το στιγμιότυπο μίας κλάσης, δηλαδή αυτή καθαυτή η δομή δεδομένων βασισμένη στο «καλούπι» που προσφέρει η κλάση. Παραδείγματος χάρη, σε μία αντικειμενοστραφή γλώσσα προγραμματισμού θα μπορούσαμε να ορίσουμε κάποια κλάση ονόματι animals, η οποία αναπαριστά το ζωικό βασίλειο, και να δηλώσουμε ένα αντικείμενο της με όνομα lion. Το αντικείμενο αυτό θα έχει δεσμεύσει χώρο στη μνήμη με βάση τις μεταβλητές και τις μεθόδους που περιγράψαμε όταν δηλώσαμε την κλάση. Έτσι, στο αντικείμενο θα μπορούσε να περιέχεται ένα γνώρισμα Country (=υπόλοιπο) και μία μέθοδος getCountry (=επέστρεψε το χώρα).

```
class animas{  
  
var lion = new animals()  
  
function getCountry();  
  
}
```

Ακολούθως θα μπορούσαμε να δημιουργήσουμε ακόμα ένα ή περισσότερα αντικείμενα της ίδιας κλάσης τα οποία θα είναι διαφορετικές δομές δεδομένων (διαφορετικά ζώα στο παράδειγμα). Ας σημειωθεί εδώ πως τα αντικείμενα μίας κλάσης μπορούν να προσπελάσουν τα ιδιωτικά περιεχόμενα άλλων αντικειμένων της ίδιας κλάσης.

4.3 HTML

Η HTML (HyperText Markup Language) είναι η κύρια γλώσσα σήμανσης για ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από *ετικέτες*, οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται *ετικέτα έναρξης* και τη δεύτερη *ετικέτα λήξης* (ή σε άλλες περιπτώσεις *ετικέτα ανοίγματος* και *ετικέτα κλεισίματος* αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

4.4 XML

Η γλώσσα XML αναπτύχθηκε από μια Ομάδα Εργασίας της XML κάτω από την καλή κηδεμονία του διεθνούς οργανισμού World Wide Web Consortium (W3C) το 1996. Εδραιώθηκε από τον John Bosak της Sun Microsystems με την ενεργή συμμετοχή μιας XML Ομάδας Ειδικού Ενδιαφέροντος που οργανώθηκε από τον οργανισμό W3C. Η XML που είναι τα ακρωνύμια των λέξεων Extensive Markup Language αποτελεί μια περιγραφική γλώσσα δομών δεδομένων που καλούνται XML documents και μερικώς περιγράφει τη συμπεριφορά των προγραμμάτων που επεξεργάζεται. Η XML ουσιαστικά προέρχεται από την SGML και αποτελεί μια προσαρμοσμένη έκδοσή της με πρωτεύων σκοπό την εκτενή χρησιμοποίησή της σε κατανεμημένες εφαρμογές. Τα XML documents που αναφέρθηκαν παραπάνω αποτελούνται από μονάδες αποθήκευσης που καλούνται έννοιες (entities) οι οποίες περιέχουν data (parse ή Unparsed). Τα parsed data αποτελούν από characters που είναι

είτε δεδομένα ή περιγραφικά στοιχεία (markup). Οι markup χαρακτήρες δεν είναι τίποτε άλλο από περιγραφές της διάταξης και λογικής δομής των δεδομένων που εμπεριέχει και ένα μηχανισμό ενσωμάτωσης περιορισμών (constraints).

4.5 CSS

Το CSS (Cascading Style Sheets), αποτελείται από ένα σύνολο ορισμάτων, διαφορετικό από αυτών στην HTML. Τα ορίσματα αυτά έχουν αναπτυχθεί με στόχο την καλύτερη προβολή και παρουσίαση των ιστοσελίδων. Μέσος του CSS μπορούν να ορισθούν χαρακτηριστικά γραμματοσειρών, όπως χρώματα, στοίχιση, και τα λοιπά, το background της εφαρμογής, περιγράμματα, κλπ. Ουσιαστικά αποτελούν κλάσεις αντικειμένων τα οποία παρέχουν μεθόδους για υλοποίηση σε κάποια tags της HTML.

Κεφάλαιο 5. Σχεδιασμός Εκπαιδευτικού Λογισμικού

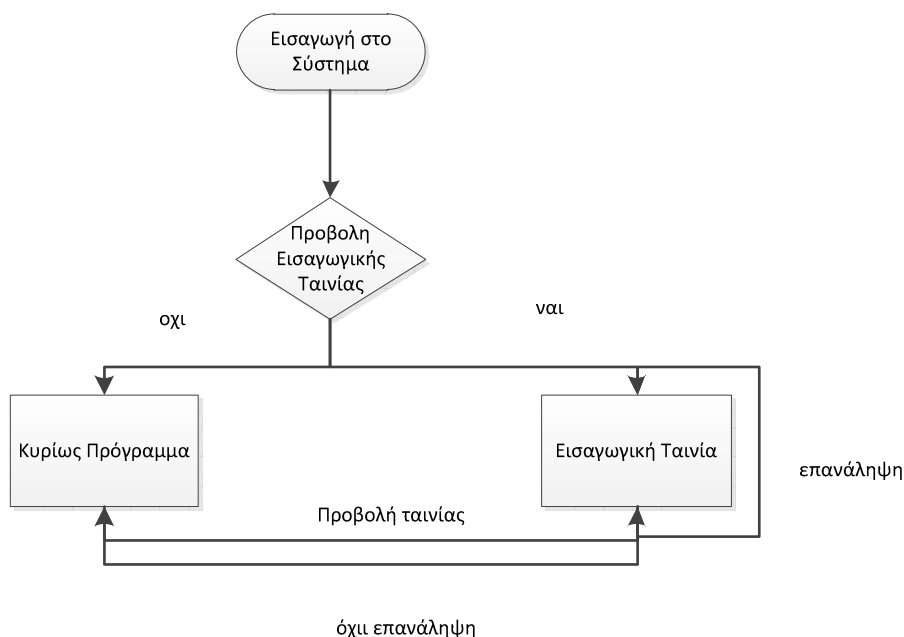
5.1 Εισαγωγή

Όπως κάθε λογισμικό, έτσι και το εκπαιδευτικό λογισμικό για το μάθημα των Δικτύων Υπολογιστών, που παρουσιάζουμε, αποτελείται από κάποιους κύκλους ανάπτυξης. Το αρχικό στάδιο αποτελείται από τη καταγραφή των βασικών λειτουργιών και στόχων που θέλουμε να επιτυγχάνει. Έπειτα, ακολουθεί ο σχεδιασμός τόσο των λειτουργιών αυτών όσο και του επιπέδου διεπαφής με τον χρήστη, και τέλος η ανάπτυξή του και τα τεστ για την εξασφάλιση της επιτυχής του λειτουργίας.

5.2 Περίληψη του Εκπαιδευτικού Λογισμικού

Μία από τις βασικές μας επιθυμίες είναι η ομαλή εισαγωγή του χρήστη στο εκπαιδευτικό πρόγραμμα. Με την έννοια ομαλή, εννοούμε μια σταδιακή προσαρμογή αντί για ένα απότομο ξεκίνημα στη διδαχθείσα ύλη. Προκειμένου να το επιτύχουμε, δημιουργήσαμε μία μικρού μήκους ταινία, σε Flash, η οποία παρουσιάζει μία περίληψη του προγράμματος και της διδαχθείσας ύλης. Η ταινία αυτή προβάλλεται σε συνδυασμό με μουσική, έτσι ώστε να εισάγουμε και διασκεδαστικά στοιχεία.

Λαμβάνοντας υπόψιν ότι αυτή η τεχνική δεν είναι αποδεκτή από όλο το ποσοστό των χρηστών, δίνουμε την επιλογή στο χρήστη για διακοπή της ταινίας και εισαγωγή του στο κυρίως κομμάτι του εκπαιδευτικού λογισμικού. Επίσης, δίνουμε τη δυνατότητα για επανάληψη της αναπαραγωγής της ταινίας, σε περίπτωση που το επιθυμεί ο χρήστης. Οι λειτουργίες αυτές παρουσιάζονται στο διάγραμμα της Εικόνας 4.2.



Εικόνα 4.2 Επιλογή Προβολής Εισαγωγικής Ταινίας

Σε κάθε περίπτωση, ο χρήστης μπορεί να επιλέξει να δει την ταινία, να επιλέξει επανάληψη της προβολής, να επιλέξει παράληψη της προβολής, αλλά και σε περίπτωση που έχει επιλέξει παράληψη και αλλάξει γνώμη, να μπορέσει να την προβάλλει.

5.3.1 Χαρακτηριστικά Διδαχθείσας Ύλης

Το λογισμικό το οποίο αναπτύξαμε αποτελεί ένα εκπαιδευτικό πρόγραμμα για τα "Δίκτυα των Υπολογιστών και Τηλεπικοινωνιών ". Ως βασικό στόχο έχουμε να διδάξουμε τόσο βασικές έννοιες, λειτουργίες και χαρακτηριστικά, όσο και πιο προχωρημένα θέματα. Με βάση αυτή την ανάγκη, το λογισμικό μας απευθύνεται σε δύο κατηγορίες χρηστών:

- Αρχάριος χρήστης: σε αυτήν την κατηγορία ανήκουν οι χρήστες οι οποίοι δεν έχουν κάποια εμπειρία με το μαθησιακό αντικείμενο. Η διδαχθείσα ύλη που αντιστοιχεί σε αυτήν την κατηγορία αποσκοπεί στο να εισάγει το χρήστη σε βασικές έννοιες, και να του δώσει μία γενική ιδέα του τομέα των Δικτύων.
- Προχωρημένος χρήστης: σε αυτήν την κατηγορία ανήκει ο χρήστης οποίος θεωρείται ότι είναι οικείος με τις βασικές έννοιες των Δικτύων, και είναι ικανός να ανταπεξέλθει σε πιο προχωρημένα θέματα.

Οι λόγοι που αποφασίσαμε να κατηγοριοποιήσουμε τους χρήστες αλλά και τη διδαχθείσα ύλη ποικίλλουν. Καταρχάς, για λόγους διαδραστικότητας. Δίνοντας στο χρήστη αυτήν την επιλογή, του δίνουμε την ελευθερία να καθορίσει ο ίδιος κατά κάποιον τρόπο, το περιεχόμενο της διδαχθείσας ύλης.

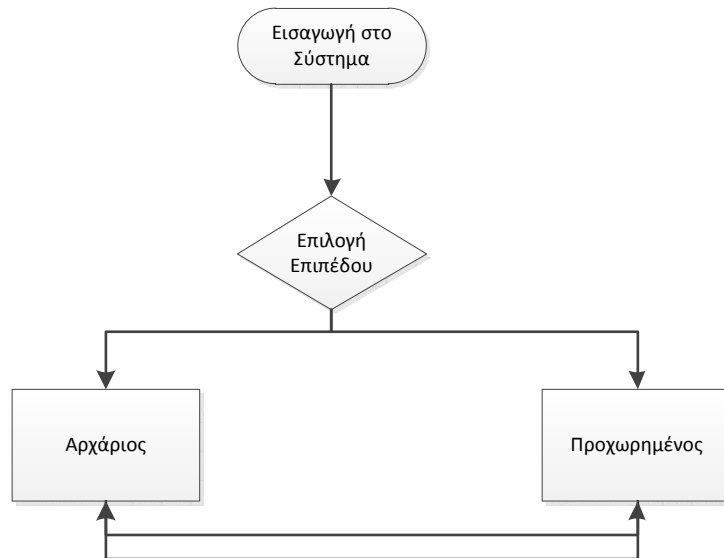
Επίσης, το λογισμικό που αναπτύξαμε, επιθυμούμε, να είναι ελκυστικό και ενδιαφέρον στο χρήστη. Στηριζόμενοι σε αυτήν την αρχή, η διδαχθείσα ύλη από μόνη της παίζει ένα ρόλο κλειδί ως προς το ενδιαφέρον του χρήστη. Έτσι, ένας πεπειραμένος στα Δίκτυα χρήστης, όχι μόνο δεν θα βρει ενδιαφέρον το εκπαιδευτικό, αλλά θα το βρει βαρετό και ανούσιο. Το ίδιο ισχύει και για τον αρχάριο χρήστη, ο οποίος εάν βρεθεί αντιμέτωπος με προχωρημένες έννοιες και προβλήματα του τομέα των Δικτύων, δεν θα έχει τα κατάλληλα εφόδια ώστε να ανταπεξέλθει στη διδαχθέν ύλη, με αποτέλεσμα να χάσει κάθε ενδιαφέρον για το πρόγραμμά.

Ένας άλλος λόγος για το διαχωρισμό των χρηστών και της ύλης είναι ότι επιθυμούμε η διδαχθέν ύλη να έχει κάποια συνέχεια. Σε πιο λεπτομέρεια, επιθυμούμε να κατηγοριοποιήσουμε το επίπεδο των εννοιών και των προβλημάτων που προβάλλονται. Με τον τρόπο αυτό, ένας αρχάριος, με τη βασική του εκπαίδευση μπορεί να προχωρήσει σε πιο προχωρημένα και λεπτομερή προβλήματα του τομέα των Δικτύων.

Τέλος, κάνοντας το διαχωρισμό αυτό, δημιουργούμε την αίσθηση στο χρήστη της προόδου (κυρίως για τους αρχάριους χρήστες). Όταν ένας χρήστης ξεκινήσει την εκπαίδευση σαν αρχάριος, και μετά συνεχίσει σαν προχωρημένος, αυτόματα του δημιουργείται η αίσθηση της προόδου, η οποία από ψυχολογικής απόψεως μπορεί να συμβάλλει και στον τομέα της επιβράβευσης.

Με βάση το διαχωρισμό που αναφέρουμε, όταν ο χρήστης εισάγεται στην εφαρμογή, του ζητείται να επιλέξει το γνωστικό του επίπεδο, όσο αναφορά τη διδαχθέν ύλη, όπως φαίνεται στην Εικόνα 4.3.1 Παρόλο που ο χρήστης επιλέγει το επίπεδό του, θέλουμε να δώσουμε την ελευθερία στο χρήστη να μπορεί να πλοηγηθεί και στο επίπεδο που δεν επέλεξε αρχικά. Για

τον λόγο αυτό, το επίπεδο του αρχάριου και το επίπεδο του προχωρημένου επικοινωνούν, υπό την έννοια ότι δεν είναι δύο ασύνδετα κομμάτια του λογισμικού.



Εικόνα 4.3.1 Επιλογή Επιπέδου Χρήστη

5.3.2 Πολυμεσικό Υλικό

Το εκπαιδευτικό λογισμικό που αναπτύξαμε θέλουμε να ακολουθεί τις βασικές αρχές για ένα ελκυστικό λογισμικό, που αναφέραμε στα προηγούμενα κεφάλαια. Για τον λόγο αυτόν, προβάλλουμε κομμάτια της διδαχθείσας ύλης, κάνοντας χρήση πολυμεσικών εργαλείων. Σε πιο λεπτομέρεια, στο λογισμικό που αναπτύξαμε, ο χρήστης μπορεί να βρει ήχο, εικόνα και βίντεο.

Ηχος

A. Μουσική :

Κατά τη διάρκεια πλοήγησής του στο εκπαιδευτικό λογισμικό, ο χρήστης μπορεί να επιλέξει την αναπαραγωγή μουσικής. Η μουσική αυτή αντιστοιχεί σε απαλούς ήχους, χωρίς έντονα στοιχεία έτσι ώστε να μην αποσπάται η προσοχή του. Ο λόγος που επιλέξαμε την αναπαραγωγή μουσικής κατά τη διάρκεια πλοήγησης στο λογισμικό, είναι ότι πολλές έρευνες, όπως (Standley, 1996) ,έχουν προβάλλει τα αποτελέσματα ότι η μουσική σε διάφορους τύπους ατόμων είναι ωφέλιμη ως προς της συγκέντρωση του εκπαιδευόμενου και στην καλύτερη απορρόφηση της διδαχθείσας ύλης.

Σε κάθε περίπτωση, ο χρήστης είναι αυτός που θα επιλέξει εάν θα ακούσει μουσική, τότε θα γίνει η αναπαραγωγή και τότε θα σταματήσει.

B. Εκφώνηση της διδαχθείσας ύλης :

Οι μαθησιακές ικανότητες κάθε εκπαιδευόμενου διαφέρουν, για παράδειγμα κάποιος μπορεί να έχει έφεση στο να απορροφά καλύτερα τη διδαχθείσα ύλη μέσω του διαβάσματος. Σε αντίθεση κάποιος άλλος εκπαιδευόμενος μπορεί να αποδίδει καλύτερα εάν η διδαχθείσα ύλη παρουσιάζεται με τη μορφή διάλεξης, δηλαδή ακούει τη διδαχθείσα ύλη. Λαμβάνοντας υπόψιν αυτή τη διαφοροποίηση μεταξύ των εκπαιδευόμενων, σε κάθε θεματική μας ενότητα δίνουμε τη δυνατότητα στο χρήστη να ακούσει τη διδαχθείσα ύλη, μέσω ηχογραφημένων αρχείων.

Εικόνα

Για την καλύτερη αναπαράσταση της γνώσης που θέλουμε να μεταφέρουμε στο χρήστη, κάνουμε εκτεταμένη χρήση εικόνων. Οι εικόνες αυτές δεν αντικαταστούν το κείμενο, αλλά αναπαριστούν και περιγράφουν έννοιες και προβλήματα τα οποία αναφέρονται μέσα στο κείμενο.

Βίντεο

Μετά από εκτεταμένη έρευνα, εντοπίσαμε εκπαιδευτικά βίντεο τα οποία αναπαριστούν ιδιότητες και χαρακτηριστικά των Δικτύων, τα οποία αναφέρουμε και στο κείμενο. Με την προβολή βίντεο, ο χρήστης έχει τη δυνατότητα να δει στην "πράξη" τα χαρακτηριστικά τα οποία έχει διδαχθεί. Έτσι η εκπαίδευση γίνεται πιο ωφέλιμη και αποτελεσματική. Σε κάθε περίπτωση, ο χρήστης έχει την ελευθερία να επιλέξει την προβολή του βίντεο.

5.3.3 Παιχνίδια

Βασισμένοι στις έρευνες που έχουν έρθει στη δημοσιότητα, σύμφωνα με τις οποίες τα παιχνίδια παίζουν καθοριστικό ρόλο στην εκπαίδευση, για το εκπαιδευτικό λογισμικό μας αναπτύξαμε εκπαιδευτικά παιχνίδια. Ο λόγος που επιλέξαμε την ανάπτυξή τους, είναι ότι θέλαμε να προσφέρουμε μία εναλλακτική μορφή εκπαίδευσης η οποία θα προσφέρει διασκέδαση παράλληλα με την εκπαίδευση.

5.3.4 Τεστ

Όπως και στην εκπαίδευση που λαμβάνει χώρα σε αίθουσες, αποφασίσαμε να αναπτύξουμε τεστ τα οποία να αξιολογούν την πρόοδο των εκπαιδευόμενων. Τα τεστ αυτά βασίζονται

στην διδαχθέντα ύλη που έχουμε συμπεριλάβει στο εκπαιδευτικό μας λογισμικό. Ο εκπαιδευόμενος ζητείται να απαντήσει σε πολλαπλών απαντήσεων ερωτήσεις, καθώς μετά την απάντησή του επιδεικνύεται εάν απάντησε σωστά ή λανθασμένα. Κατά τη διάρκεια του τεστ, κρατάτε η βαθμολογία του εκπαιδευόμενου, η οποία παρουσιάζεται στο τέλος του τεστ.

5.3.5 Κουίζ

Παράλληλα με τα τεστ που αναφέραμε πριν, αναπτύξαμε και κουίζ τα οποία βασίζονται εξίσου στη διδαχθείσα ύλη. Η διαφοροποίηση των τεστ και των κουίζ γίνεται στη διαδραστικότητα και στα πολυμέσα που χρησιμοποιούνται. Τα κουίζ που αναπτύξαμε παρουσιάζουν μία πιο διασκεδαστική μορφή, για τον εκπαιδευόμενο, αξιόλογη γνώσεων. Σε πιο λεπτομέρεια, τα κουίζ που αναπτύξαμε αποτελούνται από διαδραστικές εφαρμογές κατά τις οποίες παρουσιάζονται εικόνες στο χρήστη τις οποίες πρέπει να ταιριάζει με τις αντίστοιχες λεζάντες που περιγράφουν το περιεχόμενο των εικόνων.

5.4 Επικοινωνία

Λαμβάνοντας υπόψιν την πιθανή επιθυμία του χρήστη για επικοινωνία με τον εκπαιδευόμενο, δημιουργήσαμε ειδικές φόρμες επικοινωνίας, οι οποίες επιτρέπουν την αποστολή κειμένου στον εκπαιδευτή.⁹

⁹ Στη δική μας περίπτωση, οι φόρμες επικοινωνίας αποστέλλονται σε προσωπικό email.

5.5 Χαρακτηριστικά Λογισμικού

Όπως προαναφέραμε, η ανάπτυξη του λογισμικού έχει γίνει εξολοκλήρου βασισμένη στην τεχνολογία Flash. Εκμεταλλευόμενη τα χαρακτηριστικά της τεχνολογίας αυτής, το λογισμικό που αναπτύξαμε παρουσιάζει τα ακόλουθα χαρακτηριστικά:

- Τα κουμπιά αποτελούν κινούμενες αναπαραστάσεις.
- Η εισαγωγική ταινία μικρού μήκους, συνδυάζει εντυπωσιακά οπτικά εφέ, εικόνες και κείμενο με ήχο.
- Ενσωματωμένους players τόσο για την αναπαραγωγή μουσικής, όσο και για την αναπαραγωγή βίντεο.
- Κρυμμένο κείμενο -- το κείμενο εμφανίζεται εφόσον ο χρήστης το επιλέξει.

Κεφάλαιο 6. Υλοποίηση Εκπαιδευτικού Λογισμικού

6.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα περιγράψουμε τη διαδικασία που ακολουθήσαμε για την ανάπτυξη, καθώς και τον τρόπο λειτουργίας και τα χαρακτηριστικά του εκπαιδευτικού λογισμικού για τα Δίκτυα Υπολογιστών. Τα υποκεφάλαια χωρίζονται σύμφωνα με τις ενότητες του λογισμικού και τις περαιτέρω εφαρμογές που έχουν αναπτυχθεί.

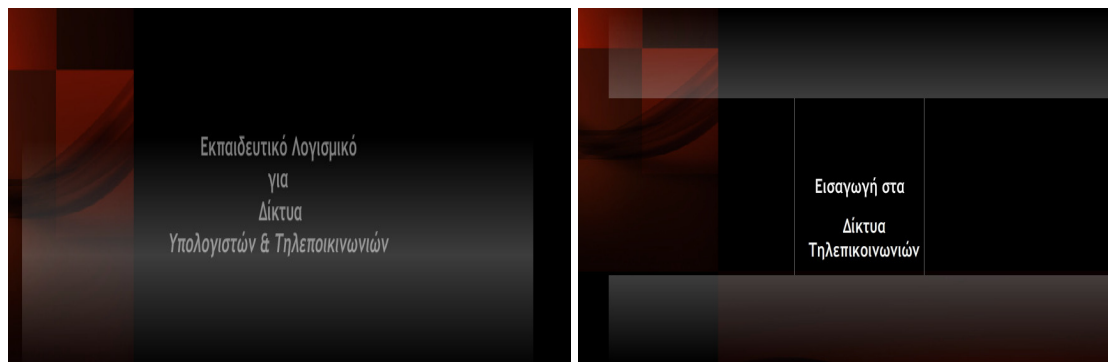
6.2 Υπό-εφαρμογές

Σε αυτό το κεφάλαιο θα περιγράψουμε τη λειτουργικότητα και τη διαδικασία ανάπτυξης των διαφόρων εφαρμογών που χρησιμοποιούμε στο εκπαιδευτικό μας λογισμικό. Στο επόμενο

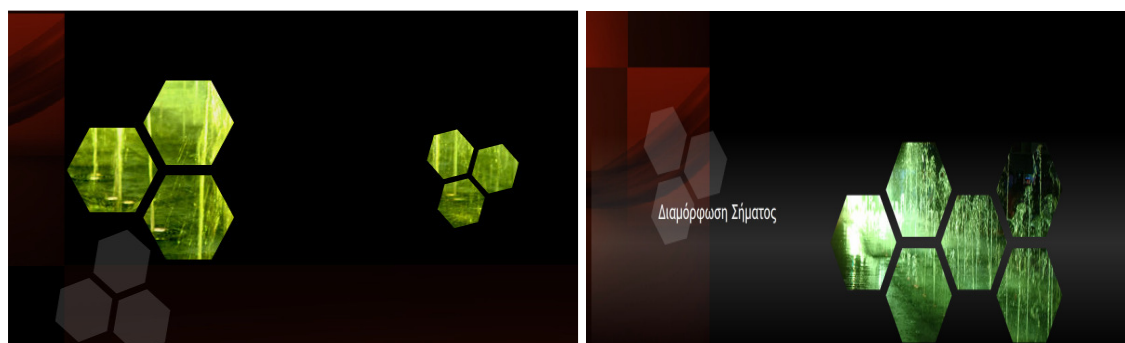
κεφάλαιο θα συζητήσουμε πως επιτυγχάνεται η ένωση αυτών των εφαρμογών σε ένα ενιαίο λογισμικό.

6.3.1 Intro

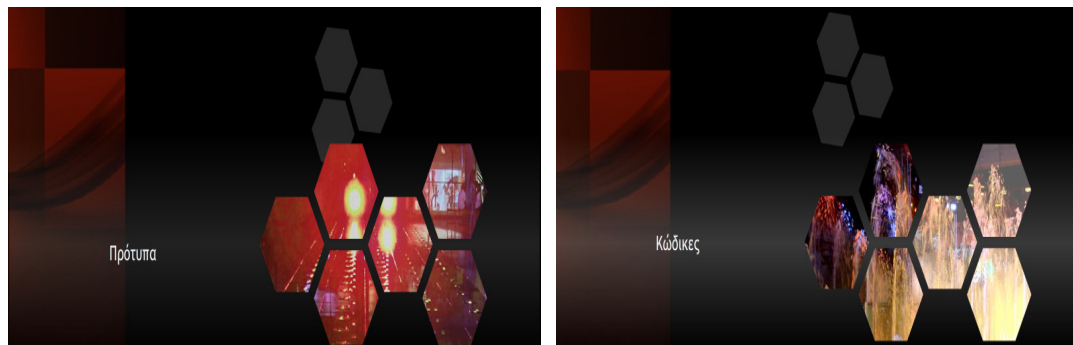
Κατά την είσοδό του στο λογισμικό, παρουσιάζεται στον χρήστη μία εισαγωγική ταινία σε flash, η οποία παρουσιάζει μία περίληψη του λογισμικού. Στις εικόνες 6.1, 6.2, 6.3, 6.4 παρουσιάζουμε κάποια στιγμιότυπα από την εισαγωγική ταινία. Η ταινία αποτελεί μία περίληψη του σκοπού του λογισμικού (Εικόνα 6.1), μία παρουσίαση των εκπαιδευτικών ενοτήτων (Εικόνα 6.2, 6.3), καθώς και τα πνευματικά δικαιώματα του λογισμικού (Εικόνα 6.4). Την ταινία τη συνοδεύει μουσική, καθώς τα γραφικά και οι τίτλοι είναι κινούμενα.



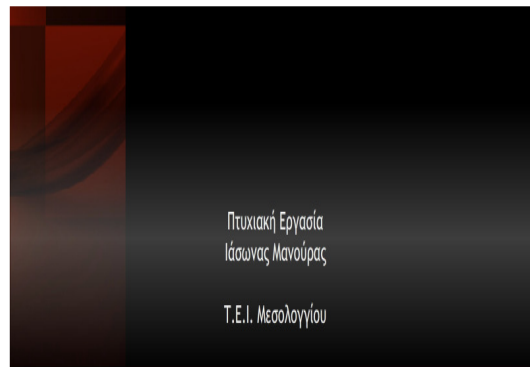
Εικόνα 6.1



Εικόνα 6.2

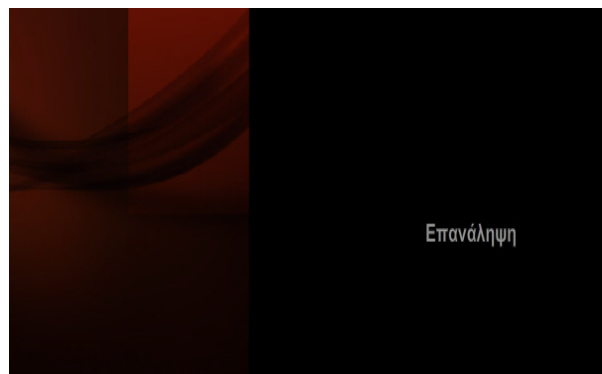


Εικόνα 6.3



Εικόνα 6.4

Μετά την ολοκλήρωση της ταινίας, εμφανίζεται στο χρήστη η επιλογή για επανάληψη της ταινίας, Εικόνα 6.5

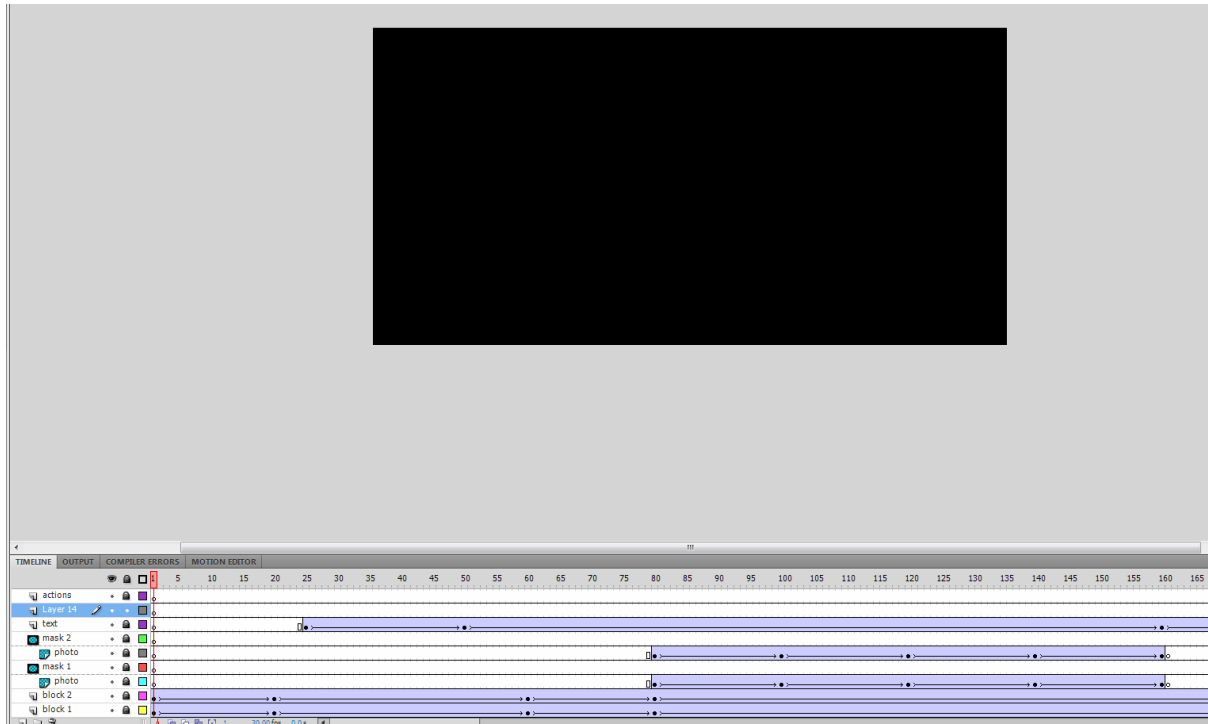


Εικόνα 6.5

6.3.2 Ανάπτυξη

Η ανάπτυξη της ταινίας αυτής έχει γίνει εξολοκλήρου σε flash, κάνοντας χρήση πολυμεσικού υλικού όπως εικόνα και ήχος. Το προγραμματιστικό περιβάλλον για τη δημιουργία της εικόνας παρουσιάζεται στην Εικόνα 6.6 Στο κάτω μέρος βρίσκεται το Timeline, το οποίο

παρουσιάζει τη χρονική εξέλιξη οποιουδήποτε animation θέλουμε να υλοποιήσουμε στο Flash. Το timeline παρουσιάζει το σύνολο των καρτέ/frames και όχι το σύνολο του χρόνου που χρειάζεται για την υλοποίηση ενός animation.



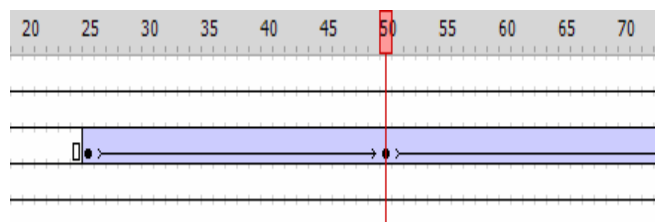
Εικόνα 6.6

Έχουμε χωρίσει την ταινία σε καρτέ, σε κάθε ένα από το οποίο έχουμε τοποθετήσει διαφορετικό πολυμεσικό υλικό.

Τίτλος

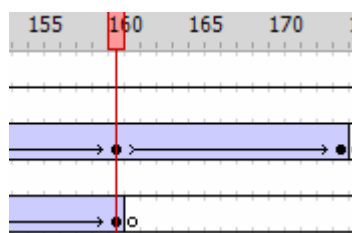
Στην αρχή της ταινίας παρουσιάζεται ο τίτλος του λογισμικού. Προκειμένου να δώσουμε μία ομαλή κίνηση στον τίτλο, αντί να χρησιμοποιήσουμε το κλασσικό εργαλείο δημιουργίας κειμένου, αρχικά δημιουργήσαμε το αντίστοιχο κείμενο, το οποίο στη συνέχεια το μετατρέψαμε σε movie clip αντικείμενο. Θέλουμε ο τίτλος αυτός να εμφανίζεται από το frame 25 έως το frame 173, καθώς η κίνησή του να είναι από την κορυφή της σελίδας μέχρι

τη μέση. Για τον λόγο αυτό τοποθετήσαμε το movie clip τίτλο στο frame 25, στη θέση στην οποία θέλουμε να ξεκινήσει η κίνηση, στην κορυφή της εφαρμογής. Έπειτα επιλέξαμε το frame 50 και μεταφέραμε το movie clip τίτλο στη μέση της εφαρμογής. Προκειμένου η κίνηση αυτή να είναι ομαλή, επιλέξαμε τα καρτέ 25-173, και στη συνέχεια επιλέξαμε την επιλογή tween. Καθότι η αλλαγή θέσεως του movie clip λαμβάνει χώρα μόνο στο frame 50, το tween στοχεύει στην ομαλοποίηση της κίνησης μέχρι αυτό το frame, Εικόνα 6.7



Εικόνα 6.7

Στη συνέχεια, για τα frames 165-170 επιλέξαμε το clip να γίνεται fade out (να εξαφανίζεται σταδιακά), με στόχο να είναι αόρατο στο frame 170. Η σταδιακή εξαφάνιση γίνεται με την επιλογή tween σε αυτά τα frames, Εικόνα 6.8



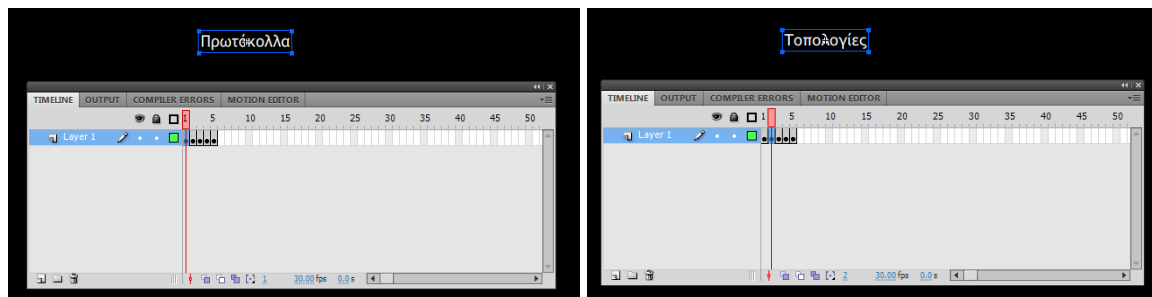
Εικόνα 6.8

Κείμενα

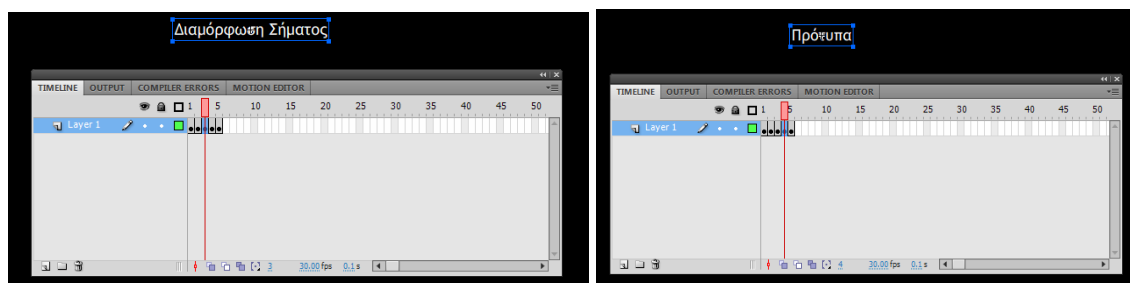
Όλα τα κείμενα δημιουργήθηκαν με την ίδια διαδικασία που περιγράψαμε για τον τίτλο. Δηλαδή, αντί για τη δημιουργία κλασικών κειμένων, επιλέξαμε τη δημιουργία movie clips,

προκειμένου να μπορούμε να επεξεργαστούμε καλύτερα την κίνησή τους. Με αυτόν τον τρόπο μπορούμε να κάνουμε χρήση των tweens, έτσι ώστε να δημιουργούμε ροή στην κίνηση.

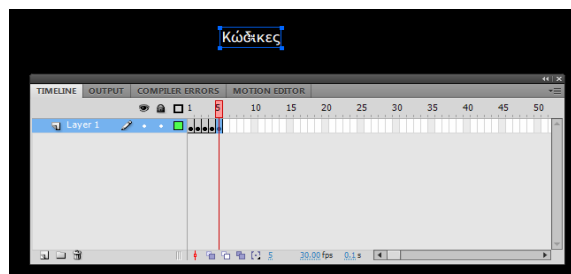
Εκτός από τα movie clips τα οποία περιλαμβάνουν απλά κείμενα, δημιουργήσαμε και ένα movie clip το οποίο περιλαμβάνει εναλλασσόμενο κείμενο. Χωρίσαμε το timeline movie clip σε εναλλασσόμενα κείμενα. Σε πιο λεπτομέρεια, σε κάθε frame του movie clip εμφανίζεται και ένα διαφορετικό κείμενο, Εικόνες 6.9 - 6.11 .



Εικόνα 6.9



Εικόνα 6.10

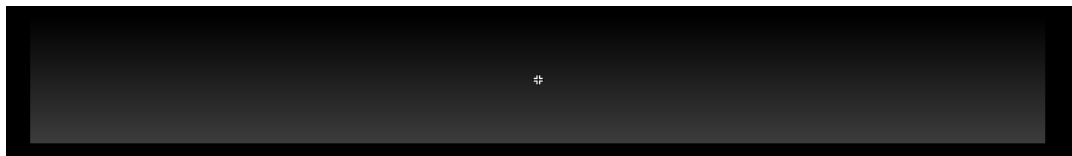


Εικόνα 6.11

Με αυτόν τον τρόπο επιτυγχάνουμε την εμφάνιση πολλαπλών κειμένων στο ίδιο frame του κυρίως timeline, καθώς η χρονική εναλλαγή τους αποτελείται από σταθερή περίοδο.

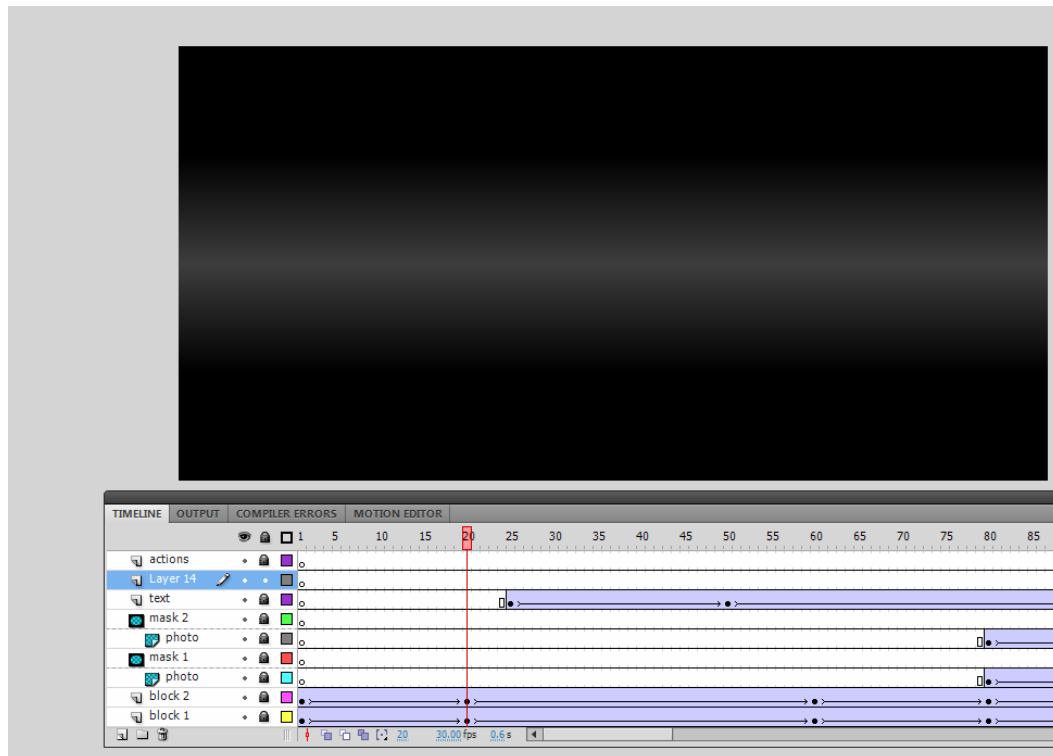
Background

Κατά την έναρξη της ταινίας, στο φόντο του τίτλου έχουμε δημιουργήσει ένα κινούμενο φόντο με τη χρήση εικόνων. Σε πιο λεπτομέρεια, δημιουργήσαμε ένα movie clip από τη εικόνα που παρουσιάζεται στη Εικόνα 6.12.



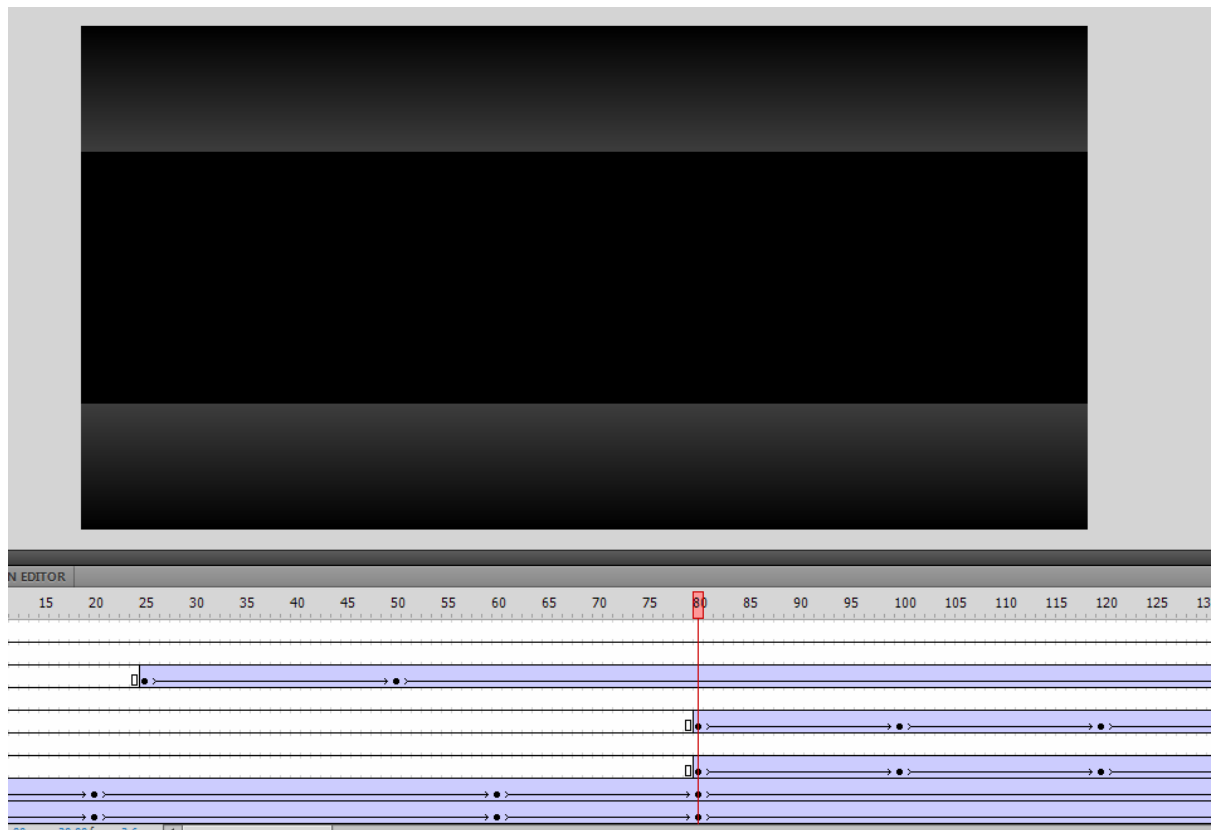
Εικόνα 6.12

Στη συνέχεια, επιλέξαμε να εμφανίζεται σταδιακά κατά την έναρξη της ταινίας (blur), με στόχο την πλήρη εμφάνισή του στο frame 20, Εικόνα 6.13. Αυτό το καταφέραμε, επιλέγοντας το επίπεδο blur στο frame 0 να είναι 70, ενώ στο frame 20 να είναι 0. Στη συνέχεια, επιλέξαμε τα frames 0-20 και κάναμε tween, προκειμένου το blur να γίνεται σταδιακά.



Εικόνα 6.13

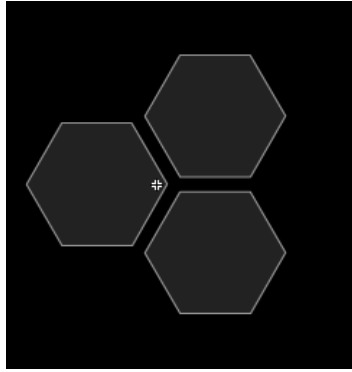
Έπειτα, κάναμε duplicate το movie clip αυτό, και ενώσαμε τον "κλώνο" του στο πάνω μέρος του αρχικού (frame 20). Επιλέγοντας σαν αρχική θέση (frame 20) των δύο αυτών clips τη μέση της ταινίας, σκοπός μας είναι τα δύο αυτά clips να αρχίσουν να απομακρύνονται σταδιακά το ένα από το άλλο, προς αντίθετες κατευθύνσεις, με τελική θέση (frame 80) το πάνω και το κάτω μέρος της ταινίας. Προκειμένου να το επιτύχουμε αυτό, επιλέξαμε το frame 80, και τοποθετήσαμε τα δύο movie clips στην επιθυμητή θέση. Στη συνέχεια επιλέξαμε tween για τα frames 20-80, Εικόνα 6.14.



Εικόνα 6.14

Η διαδικασία αυτή επαναλαμβάνεται κατά τη διάρκεια όλης της ταινίας, με τα δύο αυτά monie clips να μετακινώντας από τα άκρα της ταινίας, στη μέση, και να γίνονται blur ανά τακτούς περιόδους.

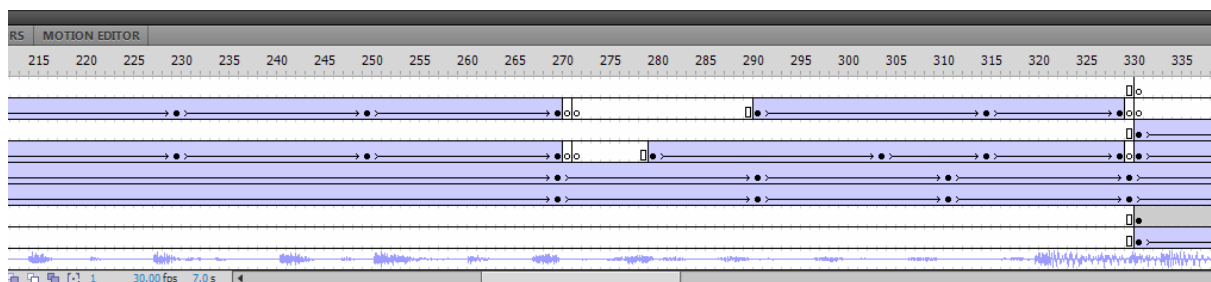
Αντίστοιχη διαδικασία ακολουθούμε και για τις υπόλοιπες κινούμενες εικόνες που χρησιμοποιούμε, όπως τα πολύγωνα, Εικόνα 6.15, στα οποία εναλλάσσουμε τις αρχικές τους θέσεις, κάνοντας παράλληλα χρήση tween προκειμένου να δημιουργήσουμε την ψευδαίσθηση της κίνησης.



Εικόνα 6.15

Ήχος

Κατά τη διάρκεια της ταινίας έχουμε επιλέξει την αναπαραγωγή μουσικής, προκειμένου να δημιουργήσουμε πιο "δραματική" αναπαραγωγή. Για την αναπαραγωγή της μουσικής δεν έχουμε δημιουργήσει κάποιο κουμπί επιλογής. Η διαδικασία που ακολουθήσαμε είναι η εξής: αρχικά κάναμε import το μουσικό κομμάτι στη βιβλιοθήκη της ταινίας μας, και στη συνέχεια τοποθετήσαμε όλο το μουσικό κομμάτι στο timeline, έτσι ώστε να εξασφαλίσουμε την επιτυχή αναπαραγωγή του από την έναρξη έως τη λήξη της ταινίας, Εικόνα 6.16.



Εικόνα 6.16

Επανάληψη

Προκειμένου ο χρήστης να μπορεί να επαναλάβει την αναπαραγωγή της ταινίας, σε περίπτωση που το επιθυμεί, δημιουργήσαμε ένα κουμπί "επανάληψη". Η διαδικασία για τη δημιουργία του είναι η εξής:

1. Αρχικά δημιουργούμε ένα κείμενο, με τη λέξη "Επανάληψη".
2. Στη συνέχεια, επιλέγουμε το κείμενο και κάνοντας δεξί κλικ και επιλέγοντας "convert to symbol" το μετατρέπουμε σε κουμπί (button).
3. Τοποθετούμε το κουμπί στο τελευταίο frame της ταινίας μας.
4. Εισάγουμε το όνομα "replay" (instance name) για το κουμπί, το οποίο θα το χρησιμοποιήσουμε στον κώδικά.
5. Εισάγουμε για το συγκεκριμένο κουμπί (στο αντίστοιχο frame), τον ακόλουθο κώδικα:

```
replay.onRelease = function()
```

```
{  
  
gotoAndPlay(1);  
  
}
```

Στο κομμάτι αυτό κώδικα, δηλώνουμε ότι εάν γίνει κάποιο κλικ σε αυτό το κουμπί, στο "release" του κουμπιού να αναπαραχθεί το frame 1 της ταινίας, το οποίο είναι και το αρχικό frame.

Περαιτέρω διευκρινήσεις

Προκειμένου η ταινία να μη αναπαράγεται επ' άπειρο, στο τελευταίο frame της ταινίας έχουμε εισάγει τον εξής κώδικα:

```
stop();  
  
stopAllSounds();
```

Με την κλήση της συνάρτησης `stop()` (η οποία είναι `build in` συνάρτηση), δηλώνουμε ότι οποιαδήποτε αναπαραγωγή `movie clip` στην ταινία σταματάει. Αντίστοιχα, με την κλήση της συνάρτησης `stopAllSounds()` οποιαδήποτε αναπαραγωγή ήχου σταματάει.

6.4.1 Τεστ

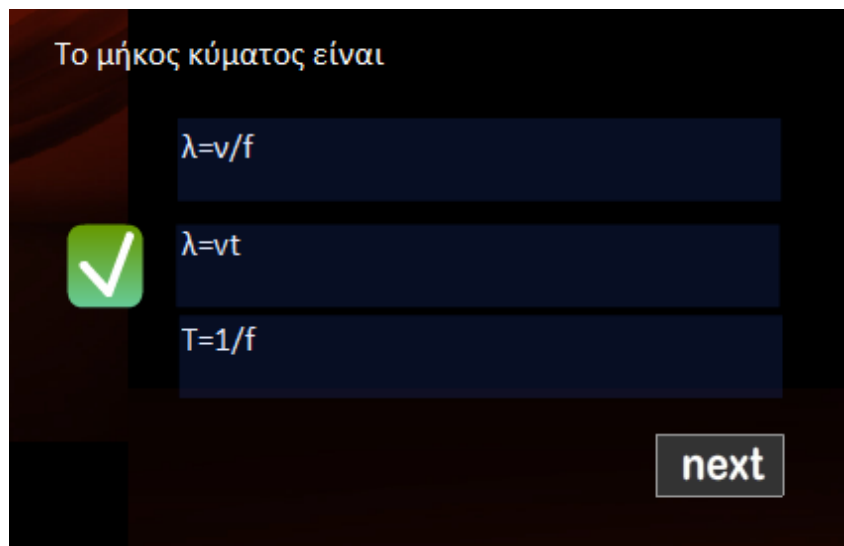
Μία από τις υπό-εφαρμογές που αναπτύξαμε για το εκπαιδευτικό λογισμικό, είναι το τεστ, το οποίο έχει διαφορετικό περιεχόμενο για το επίπεδο του "αρχάριου", και διαφορετικό περιεχόμενο για το επίπεδο του "προχωρημένου". Το τεστ είναι της μορφής πολλαπλών επιλογών, καθώς οι ερωτήσεις και οι απαντήσεις βασίζονται εξολοκλήρου στη διδακτική ύλη που έχει παρουσιαστεί στις διδακτικές ενότητες.

Τα χαρακτηριστικά του τεστ που δημιουργήσαμε είναι τα ακόλουθα:

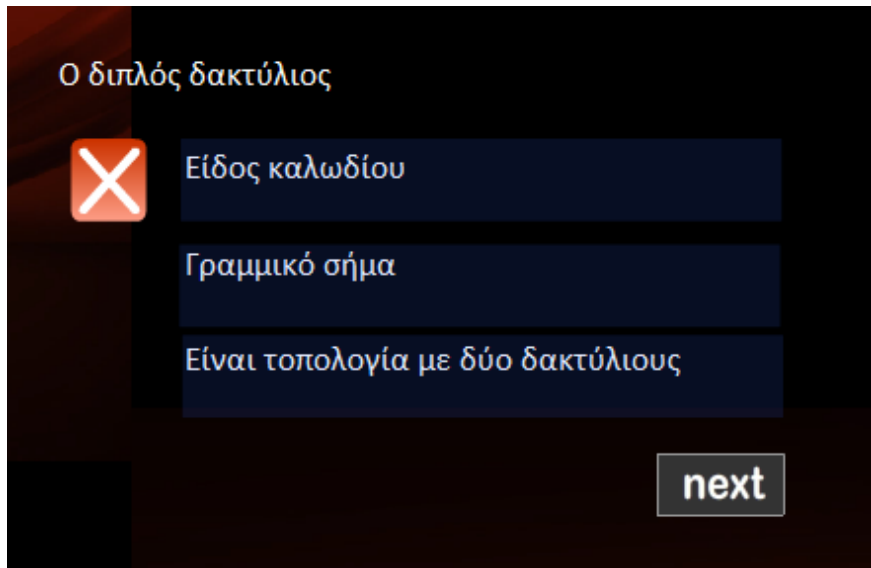
- Η σειρά εμφάνισης των ερωτήσεων να είναι τυχαία, Εικόνα 6.17
- Η σειρά εμφάνισης των απαντήσεων να είναι τυχαία.
- Να παρέχεται στο χρήστη η πληροφορία για το αν είναι σωστή ή όχι η απάντησή του, Εικόνα 6.18 - 6.19
- Να παρέχεται στο χρήστη ένα τελικό σκορ με τις σωστές και τις λανθασμένες απαντήσεις του, Εικόνα 6.20
- Να μπορούμε να επαναχρησιμοποιήσουμε το τεστ στο μέλλον, με διαφορετικές ερωτήσεις.



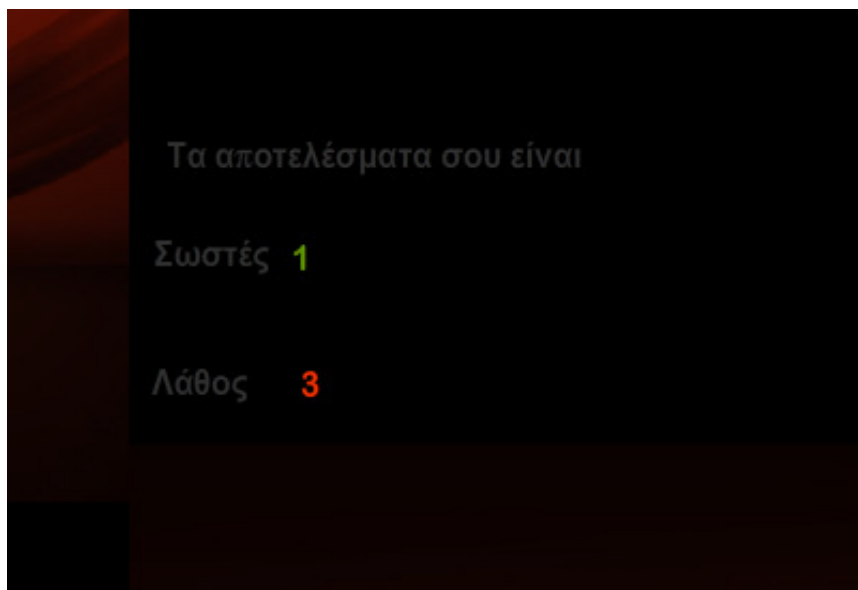
Εικόνα 6.17 Στιγμιότυπο από το Τεστ.



Εικόνα 6.18 Σωστή απάντηση στο Τεστ.



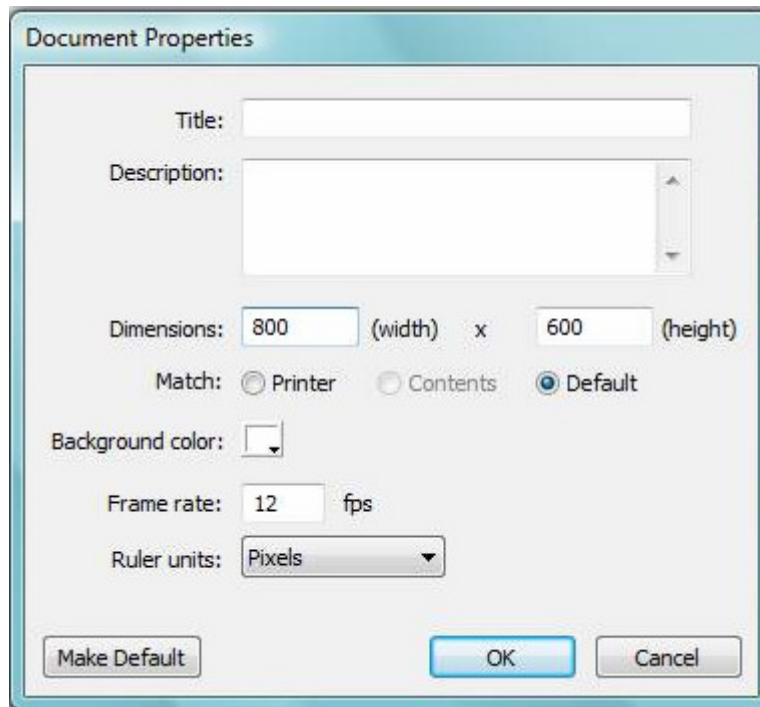
Εικόνα 6.19 Λάθος απάντηση στο Τεστ.



Εικόνα 6.20 Τελικό σκορ.

6.4.2 Ανάπτυξη

Για την ανάπτυξη της εφαρμογής Τεστ, αρχικά δημιουργήσαμε τη σκηνή πάνω στην οποία θα δουλέψουμε, Εικόνα 6.21



Εικόνα 6.21 Δημιουργία σκηνής.

Στη συνέχεια πρέπει να δημιουργήσουμε μία βάση όπου θα αποθηκεύσουμε τις ερωτήσεις και τις αντίστοιχες απαντήσεις. Οι επιλογές μας σε αυτό το σημείο είναι δύο: α) να αποθηκεύσουμε τα δεδομένα μας σε εξωτερικά XML αρχεία, β) να αποθηκεύσουμε τα δεδομένα μέσα στο ActionScript κώδικα. Αποφασίσαμε να ακολουθήσουμε τη β) επιλογή, καθώς τα δεδομένα μας (ερωτήσεις-απαντήσεις) είναι λίγα σε πλήθος και μικρά σε όγκο δεδομένων, οπότε αφενός μπορούμε να τα συμπεριλάβουμε στον κώδικα, και αφετέρου δεν επιβαρύνουμε το συνολικό πρόγραμμα με επιπλέον εξωτερικά αρχεία.

Ο τρόπος που ακολουθήσαμε για την αποθήκευση των ερωτήσεων και των απαντήσεων είναι με να δημιουργήσουμε arrays μέσα στα οποία θα τα αποθηκεύσουμε. Πιο συγκεκριμένα, με τη χρήση arrays μπορούμε να δημιουργήσουμε ένα array "questions"

```
var questions: Array["Το μήκος κύματος είναι", "Το ομοαξονικό καλώδιο", "..."]
```

και στη συνέχεια να αναφερόμαστε στα δεδομένα ως `question[0]`, `question[1]`¹⁰. Ακόλουθη τακτική ακολουθήσαμε και για τη δημιουργία των απαντήσεων, αλλά αντί για ένα μονό array δημιουργήσαμε ένα array από arrays. Ακολουθήσαμε αυτήν τη τεχνική, προκειμένου κάθε array από arrays απαντήσεων, να αντιστοιχεί σε ένα συγκεκριμένο array με ερώτηση. Για παράδειγμα:

```
answers:Array=[ ["λ=vt","T=1/f","λ=v/f"], ["Είναι από συμπαγές χάλκινο σύρμα","χρησιμοποιείται στην τηλεφωνία","Είναι το πιο γρήγορο μέσο μετάδοσης"]]
```

Με τον τρόπο αυτό μπορούμε να αναφερθούμε στο `answers[0]` το οποίο περιέχει τις πιθανές απαντήσεις του `questions[0]`.

Αποφασίσαμε να τοποθετήσουμε όλο τον κώδικα στο κυρίως timeline, και για αυτό τον λόγο δημιουργήσαμε στο πρώτο frame ένα layer με το όνομα data, Εικόνα 6.22



Εικόνα 6.22 Layer data

Σε αυτό το layer, για το επίπεδο "αρχάριος" τοποθετήσαμε τον ακόλουθο κώδικα:

```
var questions:Array=["Το μήκος κύματος είναι ","Το ομοαξονικό καλώδιο ","Ο διπλός δακτύλιος ","Οι τηλεπικοινωνίες είναι "];  
  
var answers:Array=[ ["λ=vt","T=1/f","λ=v/f"], ["Είναι από συμπαγές χάλκινο σύρμα","χρησιμοποιείται στην τηλεφωνία","Είναι το πιο γρήγορο μέσο μετάδοσης"], ["Είναι τοπολογία με δύο δακτυλίους ","Είδος καλωδίου","Γραμμικό σήμα"], ["Αποστολή δεδομένων","Το τηλέφωνο","Κινητή τηλεφωνία"] ];
```

¹⁰ Η μέτρηση ξεκινάει πάντα από το 0 και όχι από το 1.

και για το επίπεδο "προχωρημένος" τοποθετήσαμε τον ακόλουθο:

```
var questions:Array=["Το OSI είναι ","Η διορθωση σφαλμάτων ","Η πολυπλεξία διαίρεσης συχνότητας ","Οι τηλεπικοινωνίες είναι "];
```

```
var answers:Array=[ ["Μοντέλο αναφοράς Ανοικτής Διασύνδεσης Συστημάτων", "Πρωτόκολλο Τηλεπικοινωνιών", "Τρόπος Προσπέλασης Πληροφοριών"], ["Γίνεται κατα την άφιξη των πλαισίων στον παραλήπτη", "Γίνεται με τον αλγόριθμο CRC", "Κωδικοποιεί τη δέσμη σήματος"], ["Είναι τεχνολογία για μετάδοση αναλογικών σημάτων", "Είναι τεχνολογία για μετάδοση ψηφιακών σημάτων ", "Είναι μορφή πολυπλεξίας χρόνου"], ["Αποστολή δεδομένων", "Το τηλέφωνο", "Κινητή τηλεφωνία"] ];
```

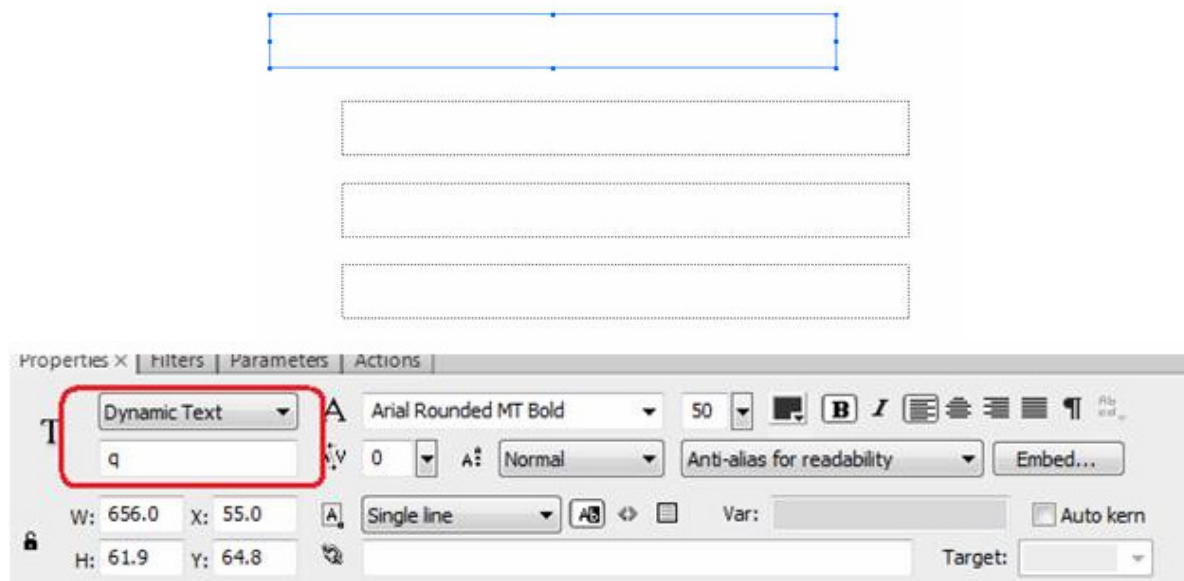
Στη συνέχεια δημιουργήσαμε άλλο ένα layer, στο κυρίως timeline, το questions, Εικόνα 6.23



Εικόνα 6.23 Questions layer

Προκειμένου να μπορέσουμε να εμφανίσουμε τις ερωτήσεις και τις απαντήσεις μας, δημιουργήσαμε ένα dynamix text¹¹ στην κυρίως σκηνή, Εικόνα 6.24

¹¹ Το κείμενο δημιουργείται δυναμικά με ActionScript κατα το χρόνο εκτέλεσης.



Εικόνα 6.24 Δημιουργία dynamic text fields

Το πρώτο field το ονομάσαμε (instance name) q (questions), ενώ τα ακόλουθα τρία, τα οποία σκοπός είναι να απεικονίζουν τις απαντήσεις, opt1, opt2, opt3(options).

Το επόμενο βήμα είναι να εξασφαλίσουμε ότι οι ερωτήσεις και οι απαντήσεις θα εμφανίζονται με τυχαία σειρά, και όχι μόνο με τη σειρά που τα έχουμε ορίσει:

```
var qno=0; var rnd1;

function change_question(){
    rnd1=Math.ceil(Math.random()*3);
    q.text=questions[qno];
    if(rnd1==1){opt1.text=answers[qno][0];opt2.text=answers[qno][1];opt3.text=answers[qno][2];}
    if(rnd1==2){opt1.text=answers[qno][2];opt2.text=answers[qno][0];opt3.text=answers[qno][1];}
    if(rnd1==3){opt1.text=answers[qno][1];opt2.text=answers[qno][2];opt3.text=answers[qno][0];}
}

change_question();
```

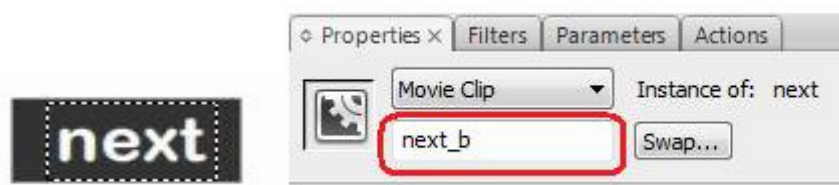
Δηλώνουμε μια καινούργια μεταβλητή rnd1. Στη συνέχεια αναθέτουμε έναν τυχαίο αριθμό μεταξύ του 1 και του 3 στο rnd1 (το Math.random()*3 είναι αυτό που θα γυρίσει τον τυχαίο αριθμό). Χρησιμοποιούμε το Math.ceil για να στρογγυλοποιήσουμε το αποτέλεσμα (σε

περίπτωση που δεν είναι στρογγυλό). Ο λόγος που αναπαράγουμε αυτόν τον τυχαίο αριθμό, είναι για τη δημιουργία μίας εκ των τριών συνδυασμών στις απαντήσεις. Με αυτόν τον τρόπο έχουμε την σωστή απάντηση να είναι opt1, opt2 ή opt3 (αλλιώς θα ήταν πάντα opt1).

Αντίστοιχα θέλουμε να εμφανίζουμε σε τυχαία σειρά τις ερωτήσεις :

```
function change_question(){  
    if(tick.visible){right_answers++;}  
    if(cross.visible){wrong_answers++;}  
    if(qno==questions.length){gotoAndPlay(2);}else{  
        tick.visible=false;cross.visible=false;  
        rnd1=Math.ceil(Math.random()*3);  
        rnd2=Math.ceil(Math.random()*questions.length)-1;  
        q.text=questions[rnd2];  
        if(questions[rnd2]=="x"){change_question();}  
        questions[rnd2]="x";  
        enable_disable(1);  
        if(rnd1==1){opt1.text=answers[rnd2][0];opt2.text=answers[rnd2][1];opt3.text=answers[rnd2][2];}  
        if(rnd1==2){opt1.text=answers[rnd2][2];opt2.text=answers[rnd2][0];opt3.text=answers[rnd2][1];}  
        if(rnd1==3){opt1.text=answers[rnd2][1];opt2.text=answers[rnd2][2];opt3.text=answers[rnd2][0];}  
    }  
}
```

Στη συνέχεια δημιουργούμε το κουμπί "next" το οποίο στοχεύει στην επόμενη ερώτηση, και δίνουμε το όνομα next_b (instance name), Εικόνα 6.25.



Εικόνα 6.25 Κουμπί next.

Έπειτα βάζουμε έναν listener¹² για να εκτελείται η συνάρτηση αλλαγής της ερώτησης με το κλικ του κουμπιού:

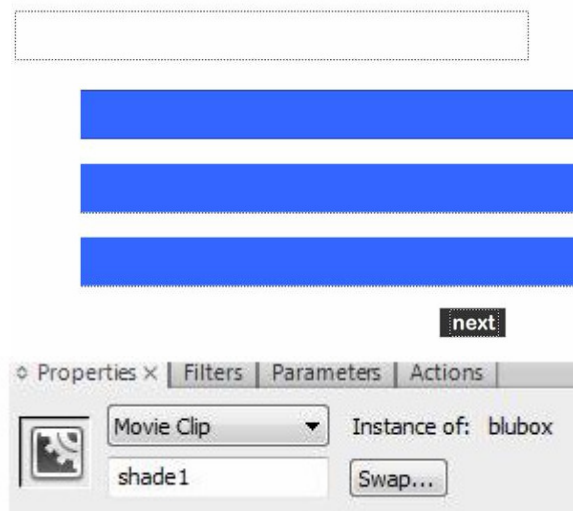
```
next_b.addEventListener(MouseEvent.CLICK, ButtonAction1);
```

Defining the function ButtonAction1:

```
function ButtonAction1(eventObject:MouseEvent) {qno++;change_question();}
```

Η συνάρτηση ButtonAction1 αυξάνει το qno κατά ένα, και στη συνέχεια καλεί την change_question.

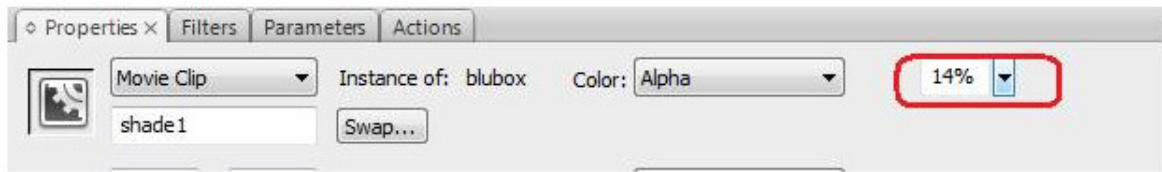
Στη συνέχεια θέλουμε να δώσουμε την δυνατότητα στο χρήστη να επιλέγει την απάντηση της αρεσκείας του. Αρχικά δημιουργούμε μπλε κουτιά πάνω από τα dynamic texts που αφορούν τις απαντήσεις, έτσι ώστε να τους δώσουμε έμφαση, και μετατρέπουμε το κάθε ένα σε movie clip (shade1, shade2, shade3), Εικόνα 6.26. Έπειτα αλλάξαμε το alpha value



Εικόνα 6.26 Δημιουργία movie clips

¹² Listeners είναι συναρτήσεις που εκτελούνται σε ανταπόκριση συγκεκριμένων events.

του καθενός σε 14% προκειμένου να φαίνεται το κείμενο (dynamic text), Εικόνα 6.27.



Εικόνα 6.27 Αλλαγή του alpha value

Στη συνέχεια προσθέτουμε ένα κομμάτι κώδικα το οποίο στην ουσία περιγράφει ότι αν ο χρήστης κάνει κλικ σε ένα από αυτά τα τρία movie clips (shade1, shade2, shade3) και η σωστή απάντηση αντιστοιχεί στην επιλογή του χρήστη, τότε εμφάνισέ το ως σωστό (τικ), αλλιώς εμφάνισέ το ως λάθος (χι).

Αρχικά δημιουργούμε την ένδειξη του σωστού και του λάθους (εικόνα με τικ, εικόνα με χι) και τα τοποθετούμε στην κυρίως σκηνή, Εικόνα 6.28.



Εικόνα 6.28 Σωστό και λάθος

Στη συνέχεια μετατρέπουμε το "χι" και το "τικ" σε movie clips δίνοντάς τους τα ονόματα "cross" και "tick" αντίστοιχα. Τα εικονίδια αυτά πρέπει να είναι αόρατα, και να εμφανίζονται αφού ο χρήστης έχει επιλέξει την αντίστοιχη απάντηση:

```
tick.visible=false;cross.visible=false;
```

Έπειτα πρέπει να εντοπίζουμε την απάντηση του χρήστη:

```
shade1.addEventListener(MouseEvent.CLICK,ButtonAction2);  
shade2.addEventListener(MouseEvent.CLICK,ButtonAction3);  
shade3.addEventListener(MouseEvent.CLICK, ButtonAction4);
```

Έπειτα εκτελούμε έναν έλεγχο για κάθε επιλογή του χρήστη: αν είναι σωστή ή όχι. Από το προηγούμενο κομμάτι κώδικα που παρουσιάσαμε, ξέρουμε ότι αν `rnd==1`, τότε `opt1` είναι το σωστό, κλ.. Με βάση αυτό, ελέγχουμε τις απαντήσεις του χρήστη:

```
functionButtonAction2(eventObject:MouseEvent)  
{if(rnd1==1){tick.visible=true;tick.y=shade1.y}else{cross.visible=true;cross.y=shade1.y}}  
functionButtonAction3(eventObject:MouseEvent)  
{if(rnd1==2){tick.visible=true;tick.y=shade2.y}else{cross.visible=true;cross.y=shade2.y}}  
functionButtonAction4(eventObject:MouseEvent)  
{if(rnd1==3){tick.visible=true;tick.y=shade3.y}else{cross.visible=true;cross.y=shade3.y}}
```

Με αυτόν τον τρόπο λέμε στο Flash να εμφανίσει το tick (visible) εάν η σωστή απάντηση είναι αυτή που έκανε κλικ ο χρήστης, αλλιώς να εμφανίσει το cross αν ο χρήστης επέλεξε τη

λάθος απάντηση. Επίσης τοποθετήσαμε τα cross/tick δίπλα από το opt που έκανε κλικ ο χρήστης. Στη συνέχεια στη change_question() πρέπει να κάνουμε τα cross/tick ξανά αόρατα:

```
tick.visible=false;cross.visible=false;
```

Το επόμενο βήμα είναι να "κλειδώνουμε" τις επιλογές του χρήστη, ώστε να μην μπορεί να τις αλλάξει:

```
function enable_disable(a){  
    if(a==0){shade1.mouseEnabled=false;shade2.mouseEnabled=false;shade3.mouseEnabled=false;}  
    if(a==1){shade1.mouseEnabled=true;shade2.mouseEnabled=true;shade3.mouseEnabled=true;}}
```

Αν καλέσουμε την enable_disable(0) θα κλειδώσει τα opt ενώ άμα καλέσουμε την την enable_disable(1) θα απενεργοποιήσει τα opt.

Για να εντοπίζουμε το πλήθος των σωστών και των λάθος απαντήσεων χρειάζεται να δημιουργήσουμε περισσότερες μεταβλητές:

```
var right_answers=0;var wrong_answers=0
```

και στη συνέχεια να προσθέσουμε έναν counter για αυτές τις μεταβλητές:

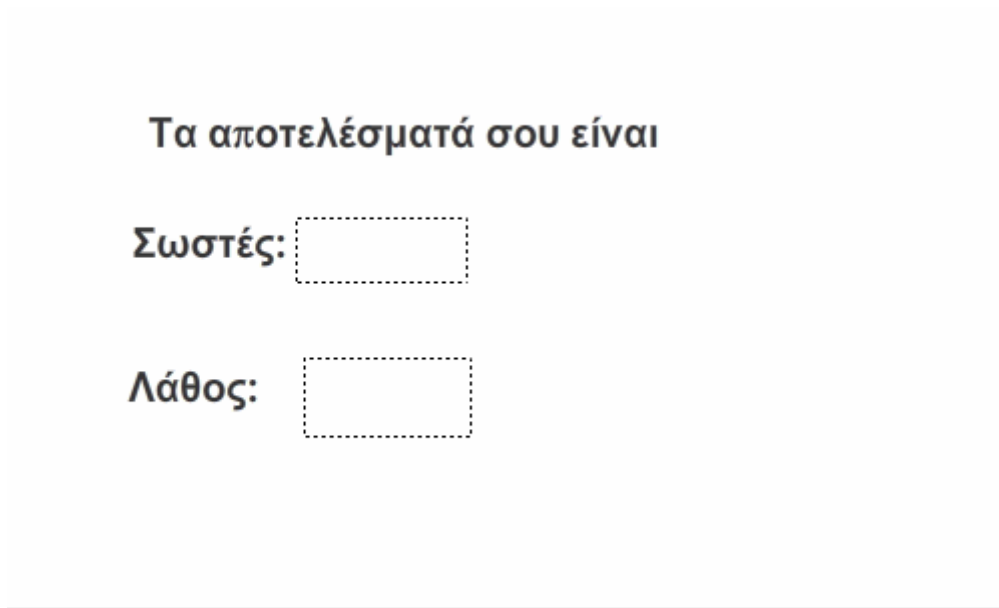
```
if(tick.visible){right_answers++;}  
    if(cross.visible){wrong_answers++;}
```

Οι ερωτήσεις θα έχουν τελειώσει όταν το qno θα είναι ίσο με το μήκος του questions (array). Όταν φτάσουμε την τελευταία ερώτηση, θέλουμε να αλλάξουμε frame στο κυρίως timeline και να προβάλλουμε το σκορ του χρήστη:

```
function change_question(){
    if(tick.visible){right_answers++;}
    if(cross.visible){wrong_answers++;}
    if(qno==questions.length){gotoAndPlay(2);}else{
        tick.visible=false;cross.visible=false;
        rnd1=Math.ceil(Math.random()*3);
        q.text=questions[qno];
        enable_disable(1);
    if(rnd1==1){opt1.text=answers[qno][0];opt2.text=answers[qno][1];opt3.text=answers[qno][2];}
    if(rnd1==2){opt1.text=answers[qno][2];opt2.text=answers[qno][0];opt3.text=answers[qno][1];}
    if(rnd1==3){opt1.text=answers[qno][1];opt2.text=answers[qno][2];opt3.text=answers[qno][0];}
    }
}
```

Στον παραπάνω κώδικα, λέμε στο Flash να πάει στο frame 2 όταν οι ερωτήσεις τελειώσουν.

Στο frame 2, τοποθετούμε τις λεζάντες που φαίνονται στη Εικόνα 6.29,



Εικόνα 6.29 Frame 2

και δημιουργούμε δύο dynamic texts: ένα για τις σωστές (ra) και ένα για τις λάθος απαντήσεις (wa). Έπειτα προσθέτουμε στον κώδικα το ακόλουθο κομμάτι :

```
ra.text=right_answers;  
wa.text=wrong_answers;
```

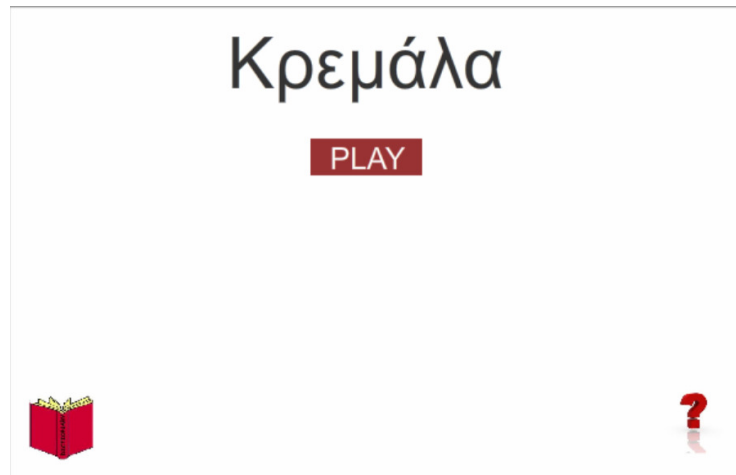
για να αντιστοιχίσουμε το κάθε dynamic text με το αντίστοιχο σκορ που θέλουμε να εμφανίζει.

6.5.1 Κρεμάλα

Βασισμένοι στις ευεργετικές ιδιότητες των παιχνιδιών στη διαδικασία εκμάθησης, και κυρίως των εκπαιδευτικών παιχνιδιών, αποφασίσαμε να αναπτύξουμε το παιχνίδι της κρεμάλας για το εκπαιδευτικό λογισμικό των Δικτύων. Σε αυτή την ενότητα θα παρουσιάσουμε τη διαδικασία που ακολουθήσαμε για την ανάπτυξη της κρεμάλας.

Το παιχνίδι της κρεμάλας είναι βασισμένο στη διδακτική ύλη που παρουσιάζουμε στο εκπαιδευτικό λογισμικό των Δικτύων. Σε πιο λεπτομέρεια, οι κρυμμένες λέξεις αφορούν ορισμούς που έχουν προαναφερθεί στις διδακτικές ενότητες. Τα χαρακτηριστικά του παιχνιδιού Κρεμάλα, που αναπτύξαμε, είναι τα εξής:

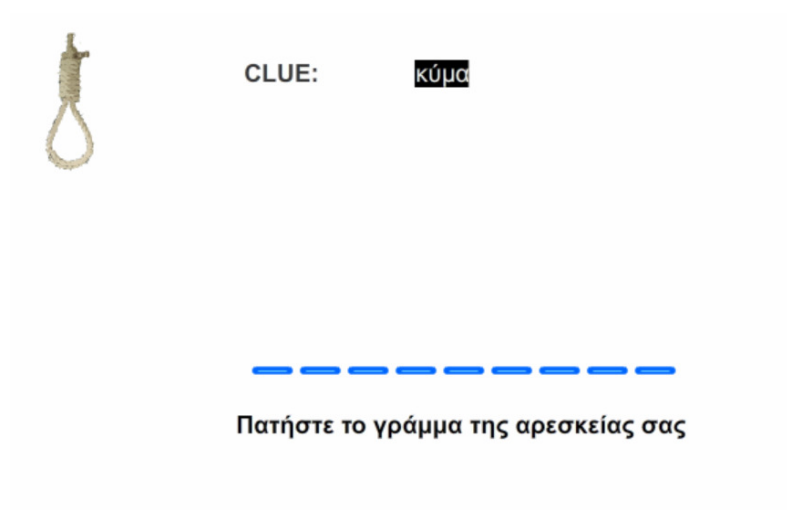
- Παροχή βοήθειας στον χρήστη, Εικόνα 6.31
- Η σειρά επιλογής των κρυμμένων λέξεων να είναι τυχαία.
- Να παρέχεται βοήθεια στο χρήστη, ως προς την κρυμμένη λέξη (clue), Εικόνα 6.32
- Να υπάρχει δυνατότητα ο χρήστης να εισάγει το γράμμα της αρεσκείας του, χρησιμοποιώντας σαν μέσο εισαγωγής το πληκτρολόγιο,
- Να παρουσιάζονται στην οθόνη τα λανθασμένα γράμματα τα οποία έχει εισάγει ο χρήστης, Εικόνα 6.33
- Να παρουσιάζονται στην οθόνη τα σωστά γράμματα τα οποία έχει εισάγει ο χρήστης, Εικόνα 6.34
- Να παρουσιάζεται στην οθόνη το πλήθος των γραμμάτων της λέξης, 6.32.
- Να παρουσιάζεται στο χρήστη η πρόοδος κατά τη διάρκεια του παιχνιδιού, υπό τη μορφή της κλασικής κρεμάλας, 6.33.
- Ο παίχτης να μπορεί να ξαναπαίξει το παιχνίδι, Εικόνα 6.35.



Εικόνα 6.30 Είσοδος στην Κρεμάλα



Εικόνα 6.31 Εμφάνιση Βοήθειας για τις Λέξεις κατά την Είσοδο στην Κρεμάλα



Εικόνα 6.32 Εμφάνιση Βοήθειας για την Κρυμμένη Λέξη –clue



CLUE: κύμα



Πατήστε το γράμμα της αρεσκείας σας

μ

Εικόνα 6.33 Εισαγωγή Λανθασμένου Γράμματος,
χρησιμοποιώντας το πληκτρολόγιο.



CLUE: κύμα



Πατήστε το γράμμα της αρεσκείας σας

μ θ λ

Εικόνα 6.34 Εισαγωγή Σωστού Γράμματος (v),
χρησιμοποιώντας το πληκτρολόγιο.



Έχασες
Πάτα οποιοδήποτε πλήκτρο για να ξαναπαίξεις

ε ζ

Πατήστε το γράμμα της αρεσκείας σας

θ υ ψ δ β ο

Εικόνα 6.35 Τελική οθόνη σε περίπτωση αποτυχίας – Δυνατότητα

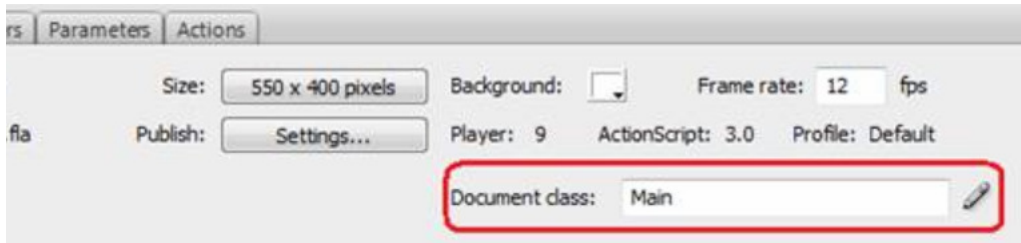
Επανάληψης του παιχνιδιού.

6.5.2 Ανάπτυξη

Για την ανάπτυξη της κρεμάλας δημιουργήσαμε δύο αρχεία: το HangMan.fla στο οποίο δουλέψαμε τα γραφικά, και το Main.as το οποίο περιέχει τον κώδικα για το παιχνίδι. Δεν υπάρχει κάποιος συγκεκριμένος λόγος που επιλέξαμε αυτή την τακτική, αντί να συμπεριλάβουμε όλον τον κώδικα στο HangMan.fla, απλώς θέλουμε να παρουσιάσουμε και αυτή την τακτική ανάπτυξης, την οποία οι περισσότεροι προγραμματιστές θεωρούν και ως πιο οργανωμένη. Προς αποφυγή επανάληψης, η ανάπτυξη του γραφικού περιβάλλοντος αναφέρεται στο HangMan.fla αρχείο ενώ ο κώδικας ActionScript αναφέρεται στο Main.as.

Το πρωταρχικό βήμα είναι η δημιουργία της κυρίως σκηνής στην οποία θα δουλέψουμε. Εφόσον επιλέξουμε τις διαστάσεις τις οποίες επιθυμούμε, το επόμενο βήμα είναι να συνδέσουμε αυτό το αρχείο με εκείνο το οποίο θα περιέχει τον κώδικα (Main.as). Αυτό το

επιτύχαμε, δηλώνοντας στα properties το όνομα του document class, Εικόνα 6.36, το οποίο πρέπει να έχει το ίδιο όνομα με το αρχείο που περιέχει τον κώδικα.

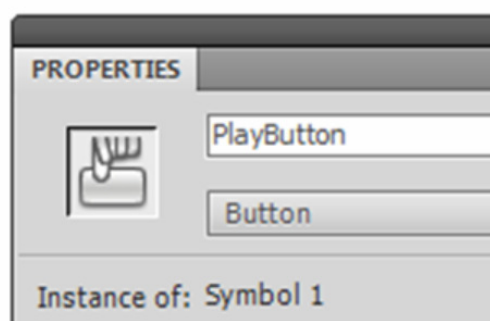


Εικόνα 6.36 Δήλωση του document class

Έπειτα προσθέτουμε στο αρχικό frame της σκηνής τα κουμπιά. Αρχικά δημιουργούμε το κουμπί εισόδου, Enter, Εικόνα 6.37, και του δίνουμε το instance name PlayButton, Εικόνα 6.38.



Εικόνα 6.37 Κουμπί Εισόδου



Εικόνα 6.38 Instance name για το Κουμπί Εισόδου

Με τον ίδιο τρόπο δημιουργούμε τα κουμπιά για τη βοήθεια, Εικόνες 6.39, 6.40.



Εικόνα 6.39 Κουμπί Βοήθειας



Εικόνα 6.40 Κουμπί Βοήθειας

Έπειτα δημιουργούμε ένα δεύτερο frame στο οποίο φτιάχνουμε το κινούμενο σχέδιο της κρεμάλας. Το κινούμενο σχέδιο της κρεμάλας το δημιουργήσαμε με τη χρήση movie clip. Σε πιο λεπτομέρεια, δημιουργήσαμε ένα movie clip με 7frames, Εικόνες 6.41 – 6.47



Εικόνα 6.41 Frame 1.



Εικόνα 6.42 Frame 2.



Εικόνα 6.43 Frame 3.



Εικόνα 6.44 Frame 4.



Εικόνα 6.45 Frame 5.



Εικόνα 6.46 Frame 6.



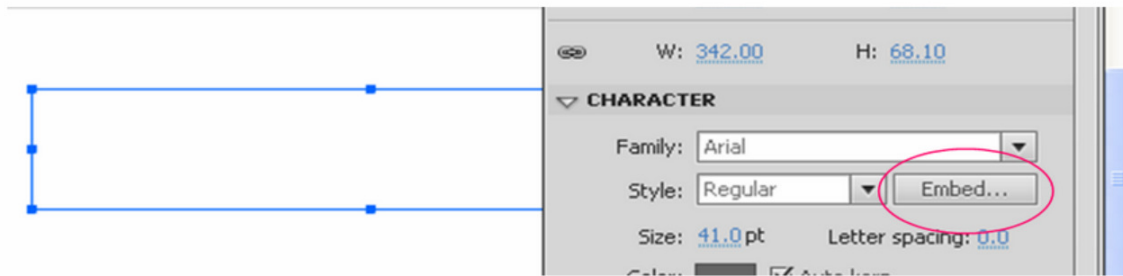
Εικόνα 6.47 Frame 7.

Στη συνέχεια ονομάσαμε το movie clip (instance name) Man. Έπειτα δημιουργήσαμε ένα στατικό κείμενο που περιέχει τη λέξη clue, και δίπλα ένα δυναμικό (dynamic text) το οποίο θα περιέχει τη βοήθεια για τη λέξη, Εικόνα 6.48.



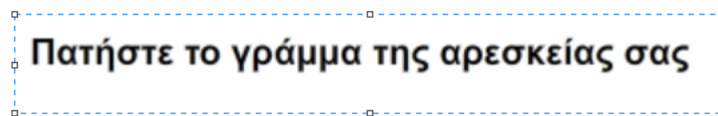
Εικόνα 6.48 Βοήθεια για τη λέξη.

Προκειμένου στο dynamic text να εμφανίζονται ελληνικά, επιλέξαμε την επιλογή Embed από τα properties του dynamic text. Στη συνέχεια τικάραμε την επιλογή Greek, Εικόνα 6.49.

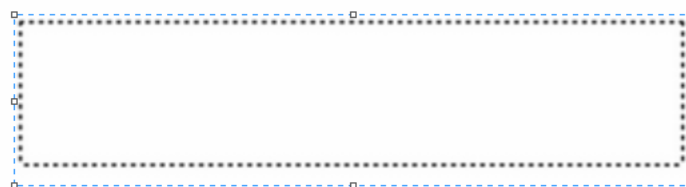


Εικόνα 6.49 Embed Properties

Στο κάτω μέρος της σελίδας δημιουργούμε ένα στατικό κείμενο το οποίο περιέχει οδηγίες προς τον χρήστη, Εικόνα 6.50, και ένα δυναμικό κείμενο στο οποίο θα εμφανίζονται οι λάθος επιλογές του χρήστη, Εικόνα 6.51.



Εικόνα 6.50 Στατικό Κείμενο



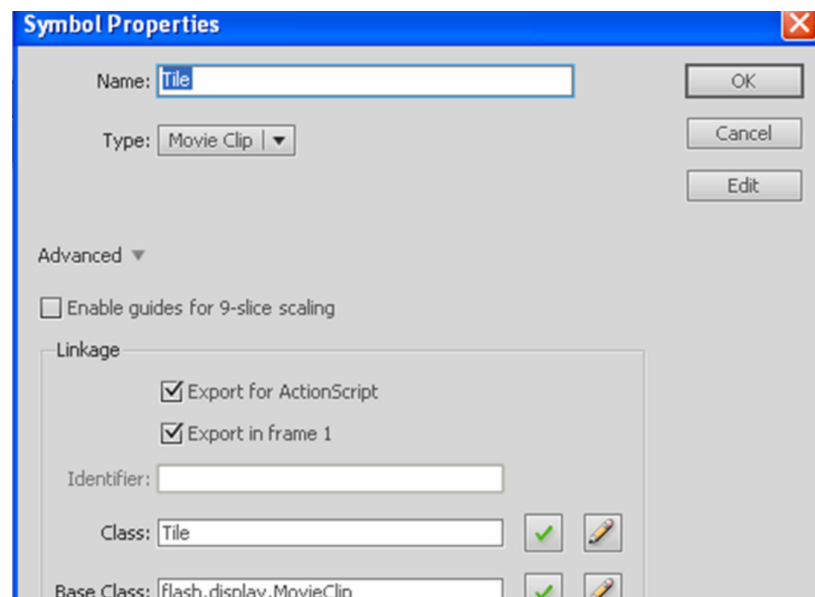
Εικόνα 6.51 Δυναμικό Κείμενο

Στη συνέχεια δημιουργούμε tiles για τα γράμματα της κρυμμένης λέξης, πάνω από τα οποία τοποθετούμε dynamic text, Εικόνα 6.52.



Εικόνα 6.52 Tile

Δημιουργήσαμε μόνο ένα tile, το οποίο με χρήση κώδικα ActionScript θα αναπαράγουμε. Προκειμένου να μπορέσουμε να εμφανίσουμε τα γράμματα της αρεσκείας μας στα tiles και να μπορέσουμε να τα χειριστούμε κατάλληλα μέσα στον κώδικα, μετατρέψαμε το tile σε movie clip, δίνοντάς του το αντίστοιχο instance name, Εικόνα 6.53.



Εικόνα 6.53 Μετατροπή tile σε movie clip

Στη συνέχεια το διαγράφουμε από την κυρίως σκηνή, και δημιουργούμε ένα movie clip το οποίο θα αποτελείται από 2 frames, κάθε ένα από τα οποία φέρει το αντίστοιχο μήνυμα, Εικόνες 6.54 - 6.55.

Εικόνα 6.54 Frame 1



Εικόνα 6.55 Frame 2

Αφού δημιουργήσαμε την κυρίως σκηνή και τα γραφικά για την κρεμάλα, το επόμενο βήμα είναι η ανάπτυξη του κώδικα προκειμένου να λειτουργεί το παιχνίδι της Κρεμάλας. Οι στόχοι που θέλουμε να επιτύχουμε μέσω της προγραμματιστικής προσέγγισης είναι οι εξής:

- Δημιουργία λίστας στην οποία θα αποθηκεύσουμε τις κρυμμένες λέξεις.
- Δημιουργία συνάρτησης η οποία θα επιλέγει τυχαία μία λέξη.
- Χώρισμα της λέξης σε γράμματα.
- Επιλογή του αντίστοιχου clue για την λέξη, και προβολή του στο χρήστη.
- Δυνατότητα για το χρήστη να εισάγει τα γράμματα χρησιμοποιώντας ως μέσο εισαγωγής το πληκτρολόγιο.
- Έλεγχος του γράμματος, το οποίο εισήγαγε ο χρήστης: εάν έχει ξαναεισαχθεί, εάν είναι σωστό ή όχι.
- Τοποθέτηση του γράμματος το οποίο εισήγαγε ο χρήστης στα tiles, εάν το γράμμα είναι σωστό, αλλιώς στο dynamic box το οποίο προβάλλει τα λάθος γράμματα.

Τα χαρακτηριστικά αυτά τα υλοποιήσαμε στην Main.as, στην οποία κάνουμε import τις κατάλληλες βιβλιοθήκες προκειμένου να μπορέσουμε να χειριστούμε κατάλληλα τα movie clips, την είσοδο από πληκτρολόγιο, και τις κινήσεις του ποντικιού.

```
import flash.display.MovieClip;

import flash.events.MouseEvent;

import flash.events.KeyboardEvent;
```

Στη συνέχεια δηλώνουμε τις μεταβλητές με τις οποίες θα δουλέψουμε.

```
Public var wordlist:WordList;

public var CurrentWord:String;
```

```
public var CurrentClue:String;

public var rnd:int;

public var Letters:Array=new Array;

public var i:int;

public var tile:Tile;

public var TileArray:Array=new Array;

public var UserKeyPress:String;

public var Alphabet:Array=new Array;

public var Found:Boolean;

public var Win:Boolean;

public var Lose:Boolean;

public var DefeatCounter:int;

public var UsedCharacters:Array=new Array;

public var FirstPlayThrough:Boolean= true;
```

Στη συνέχεια υλοποιούμε την Main function η οποία είναι η constructor και θα καλείται αυτόματα όποτε αναπαράγεται η flash εφαρμογή. Αρχικά θέλουμε να επιτύχουμε την αποφυγή αναπαραγωγής του frame 2 της flash εφαρμογής, η οποία θα επιτευχθεί κατόπιν επιλογής του χρήστη. Στη συνέχεια δημιουργούμε ένα instance της WordList, και τέλος δίνουμε λειτουργικότητα στο κουμπί Play:

```
public function Main() {

stop();

wordlist=new WordList;

PlayButton.addEventListener(MouseEvent.CLICK, ButtonAction1);
```

```
function ButtonAction1(eventObject:MouseEvent) {gotoAndStop(2); SetUp();}
}
```

Έπειτα δημιουργούμε μία ξεχωριστή class για το WordList προκειμένου να είναι εύκολη η επαναχρήση του και σε περίπτωση που θέλουμε να αυξήσουμε κατά πολύ το πλήθος των λέξεων, να γίνεται με μεγαλύτερη ευκολία μέσα στον κώδικα. Για αυτό τον λόγο δημιουργούμε ένα καινούργιο αρχείο το WordList.as, στο οποίο δηλώνουμε και τις κρυμμένες μας λέξεις:

```
package {
import flash.display.MovieClip;
public class WordList extends MovieClip {
public var Words:Array = new Array;
public var Clues:Array = new Array;
public function WordList() {
Words=["αστερας", "χερτζ", "ναυπακτος", "ραμ", "συχνοτητα", "σημα"];
Clues=["είδος τοπολογίας", "μονάδα συχνότητας", "τόπος", "μέρος υπολογιστή", "κύμα", "θόρυβος"]
}}
}
```

Όπως αναφέραμε και στην προηγούμενη ενότητα, οι θέσεις που προσδιορίζουμε για τα Words είναι συνδεδεμένες με τις λέξεις που προσδιορίζουμε για τα Clues. Έτσι το Word[0] == αστέρας, αντιστοιχεί στο Clue[0]==είδος τοπολογίας. Στη συνέχεια ορίζουμε τις συναρτήσεις και τις αντίστοιχες λειτουργίες.

Επιλογή Λέξης

Ορίζουμε τη συνάρτηση ChooseWord() στην οποία οι βασικές λειτουργίες είναι η επιλογή μίας τυχαίας λέξης από το array Words και εντοπισμός του αντίστοιχου clue, από το array Clues.

```
public function ChooseAWord(){  
    rnd=Math.random()*wordlist.Words.length;  
    CurrentWord=wordlist.Words[rnd];  
    CurrentClue=wordlist.Clues[rnd];  
    trace(CurrentWord);trace(CurrentClue);  
}
```

Αρχικά δηλώνουμε τον integer rnd, ο οποίος κυμαίνεται από το 0 έως το μήκος του array Words μείων 1 (λόγο ότι η μέτρηση της λίστας ξεκινάει από το 0). Ο rnd θα αποτελέσει τον δείκτη με βάση τον οποίο θα επιλέξουμε την λέξη και το αντίστοιχο clue.

Διαχωρισμός Γραμμάτων στις Λέξεις

Προκειμένου να χωρίσουμε την λέξη στα αντίστοιχα γράμματα, ορίζουμε την SplitIntoLetters(). Στόχος της συνάρτησης αυτής είναι να διαχωρίσει τα γράμματα τις λέξεις και να τα αποθηκεύσει στο array Letters που δηλώσαμε προηγουμένως:

```
public function SplitIntoLetters(){  
    Letters=CurrentWord.split("");  
    trace(Letters);  
}
```

Προβολή της Πληροφορίας

Ορίζουμε τη συνάρτηση Display() η οποία είναι υπεύθυνη για την προβολή τόσο των γραμμάτων της κρυμμένης λέξης, όσο του αντίστοιχου clue. Αρχικά ορίζουμε ένα MessageBoard το οποίο είναι άορατο στο χρήστη. Έπειτα, προβάλλουμε το clue στην

οθόνη στο DisplayClue κουτί (dynamic text). Τέλος, προσθέτουμε τον κατάλληλο αριθμό από tiles (βασισμένος στον αριθμό των γραμμμάτων της λέξης). Για να το καταφέρουμε, δημιουργούμε ένα νέο tile στην οθόνη, χρησιμοποιούμε τη μεταβλητή i για να χειριστούμε τη x συντεταγμένη στην οθόνη, αρχίζοντας από το σημείο 300 και για τη συντεταγμένη y το 250. Τέλος χρησιμοποιούμε ένα άλλο array, TileArray, στο οποίο θα γίνεται αναφορά για τα tiles (αποθηκεύονται τα γράμματα).

```
public function Display(){  
    MessageBoard.visible=false;  
    DisplayClue.text=CurrentClue;  
    for(i=0;i<Letters.length;i++){  
        tile=new Tile;addChild(tile);  
        tile.x=300+(35*i);tile.y=250;  
        TileArray[i]=tile;  
    }  
}
```

Έλεγχος Εισόδου

Χρειάζεται να μπορούμε να εντοπίσουμε και να διαβάσουμε την είσοδο από πληκτρολόγιο, καθώς και οποιαδήποτε κίνηση του ποντικιού (η οποία αφορά το παιχνίδι). Για τον λόγο αυτό ορίζουμε την AddControls().

Όταν ο χρήστης εισάγει κάποιο γράμμα από το πληκτρολόγιο, αυτό καταγράφεται σαν αριθμός ο οποίος αποστέλλεται στη συνάρτηση ConvertToFunction() για να μετατραπεί σε γράμμα (ASCII) και χρησιμοποιούμε το αποτέλεσμα σαν την τιμή που ανατίθεται στο UserKeyPress. Για παράδειγμα, εάν ο χρήστης εισάγει το γράμμα «α», τότε

UserKeyPress=α. Έπειτα το γράμμα αποστέλλεται στην ProcessAnswer() προκειμένου να ελεγχθεί εάν είναι σωστό ή όχι.

```
public function AddControls(){
    if(FirstPlayThrough){
        FirstPlayThrough=false;
        stage.focus=stage;
        stage.addEventListener(KeyboardEvent.KEY_UP,keyUpListener);
        function keyUpListener(e:KeyboardEvent):void {trace(e.keyCode);
            UserKeyPress=ConvertToLetter(e.keyCode);
            ProcessAnswer(UserKeyPress);
        }
    }
}
```

Μετατροπή Αριθμού σε Γράμμα

Ορίζουμε τη συνάρτηση ConvertToLetter() στην οποία δηλώνουμε το ελληνικό αλφάβητο μέσα στο array Alphabet. Γνωρίζουμε ότι ο αριθμός 64 αντιστοιχεί στο α, 65 στο β, κλπ. Εάν ο αριθμός είναι μικρότερος από 64 ή μεγαλύτερος από 89 τότε η συνάρτηση θα επιστρέψει κενό.

Χρειάζεται να μπορούμε να εντοπίσουμε και να διαβάσουμε την είσοδο από πληκτρολόγιο

```
public function ConvertToLetter(a){
    Alphabet=["α","β","γ","δ","ε","ζ","η","θ","ι","κ","λ","μ","ν","ξ","ο","π","ρ","σ","τ","υ","φ","χ","ψ","ς","ω"];
    if(a>64 && a<89){
        return(Alphabet[a-65]);
    }
    else{return("");}
}
```

Έλεγχος Απαντήσεων

Ορίζουμε τη συνάρτηση ProcessAnswers() στην οποία θα γίνεται έλεγχος της κάθε απάντησης του χρήστη. Ο έλεγχος γίνεται μόνο αν η απάντηση είναι διάφορη του κενού και εάν το παιχνίδι τρέχει ακόμα. Εάν το παιχνίδι έχει λήξει (ο χρήστης κέρδισε ή έχασε), τότε οποιοδήποτε γράμμα θα λειτουργήσει έτσι ώστε να γίνει επανεκκίνηση του παιχνιδιού, καλώντας την RestartGame().

Προκειμένου να εξετάσουμε εάν το γράμμα που εισήγαγε ο χρήστης βρίσκεται στην κρυμμένη λέξη, αρχικά υποθέτουμε ότι δεν υπάρχει θέτοντας Found=false, η οποία μεταβλητή θα γίνει true εάν το γράμμα βρεθεί. Ο έλεγχος γίνεται με όλο το array Letters. Εφόσον βρεθεί το γράμμα, ελέγχουμε εάν ο παίχτης έχει κερδίσει το παιχνίδι. Εάν το γράμμα δεν βρεθεί και δεν έχει χρησιμοποιηθεί προηγουμένως πρέπει να αλλάξουμε το γραφικό της κρεμάλας και να προβάλουμε το γράμμα αυτό στην οθόνη. Τέλος, πρέπει να ελέγξουμε εάν έχει λήξει το παιχνίδι.

```
public function ProcessAnswer(a){
    if(a!="" && !Lose && !Win){
        Found=false;
        for (i=0;i<Letters.length;i++){
            if(a==Letters[i]){Found=true;TileArray[i].Blank.text=a;}
        }
        if(Found){CheckForWin();}
        if(!Found && Unused(a)){AddBodyPart();}
        DisplayWrongLetter(a);
        CheckForDefeat();
    }
    else{
        if(Win || Lose){RestartGame();}
    }
}
```

Έλεγχος Για Νίκη

Ορίζουμε τη συνάρτηση CheckForWin(), στην οποία συγκρίνονται τα γράμματα του Tiles με τα γράμματα του Letters (τα γράμματα της λέξης με τα γράμματα του χρήστη). Άμα είναι τα ίδια, τότε προβάλουμε το μήνυμα νίκης στην οθόνη.

```

public function CheckForWin(){
Win=true;
for (i=0;i<Letters.length;i++){
if(TileArray[i].Blank.text!=Letters[i]){Win=false;}
}
if(Win){MessageBoard.visible=true;
MessageBoard.gotoAndStop(1);
}
}

```

Έλεγχος Των Γραμμάτων

Ορίζουμε τη συνάρτηση Unused() η οποία ελέγχει εάν το γράμμα που εισήγαγε ο χρήστης έχει ξαναχρησιμοποιηθεί.

```

public function Unused(a){
for (i=0;i<UsedCharacters.length;i++){
if(UsedCharacters[i]==a){return false;}
}
return true;
}

```

Αλλαγή Εικόνας της Κρεμάλας

Μέσω της συνάρτησης AddBodyPart() προχωράμε το movie clip της κρεμάλας κατά 1 frame, κάθε φορά που ο χρήστης εισάγει λάθος γράμμα.

```

public function AddBodyPart(){
Man.nextFrame();
}

```

Έλεγχος για Ήττα

Μέσω της συνάρτησης CheckForDefeat() ελέγχουμε εάν ο χρήστης έχει χάσει. Ορίζουμε έναν μετρητή, το DefeatCounter, ο οποίος είναι 6 εάν ο χρήστης έχει χάσει (επειδή ο χρήστης έχει μόνο 6 προσπάθειες). Όταν ο χρήστης έχει χάσει, Lose=true και εμφανίζεται στο MessageBoard το μήνυμα της ήττας.

```

public function DisplayWrongLetter(a){
UsedCharacters[UsedCharacters.length]=a;
BadLetters.text="";
for(i=0;i<UsedCharacters.length;i++){
BadLetters.appendText(UsedCharacters[i]+" ");
}
}

```



```
}  
}
```

Επανεκκίνηση Παιχνιδιού

Μέσω της συνάρτησης `RestartGame()` πραγματοποιείται η επανεκκίνηση του παιχνιδιού.

Στην ουσία διαγράφουμε οποιαδήποτε τιμή έχουν πάρει οι μεταβλητές μας κατά τη διάρκεια του προηγούμενου γύρου, και διαγράφουμε τα tiles από τη οθόνη.

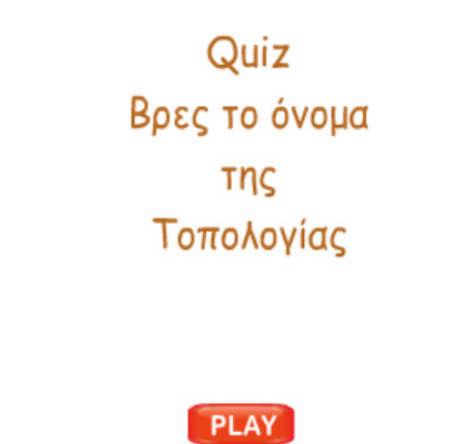
```
public function RestartGame(){  
Win=false;Lose=false;BadLetters.text="";  
for(i=0;i<Letters.length;i++){  
removeChild(TileArray[i]);TileArray[i]=null;  
}//for i  
Letters=[];UsedCharacters=[];TileArray=[];  
SetUp();  
}
```

6.6.1 Quiz

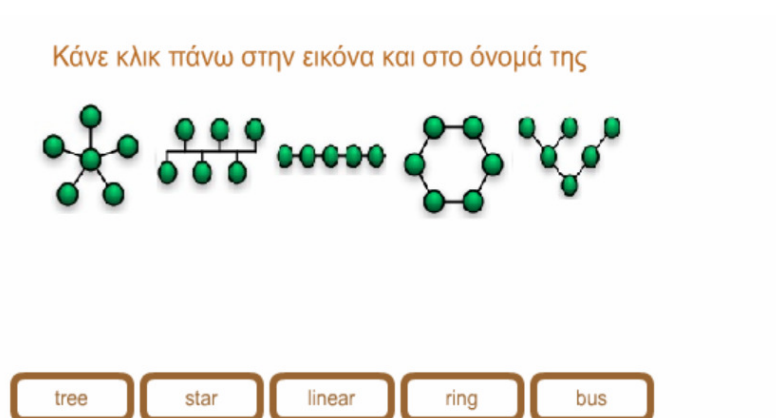
Για το εκπαιδευτικό λογισμικό των Δικτύων, αναπτύξαμε ένα Quiz κατά το οποίο ο χρήστης θα πρέπει να ταιριάζει την εικόνα που απεικονίζεται με τον τίτλο που την περιγράφει καταλληλότερα. Αυτό το είδος quiz συνδυάζει χαρακτηριστικά που συναντάμε και στα κλασικά τεστ γνώσεων, αλλά και στα παιχνίδια. Σε πιο λεπτομέρεια, απεικονίζουμε διάφορα είδη τοπολογιών καθώς και τα ονόματα των τοπολογιών αυτών σε ανακατεμένη σειρά, Εικόνα 6.57. Σκοπός είναι ο χρήστης να μπορέσει να ταιριάζει επιτυχώς κάθε τοπολογία με το όνομά της. Για το quiz αυτό χρησιμοποιήσαμε τις αγγλικές ονομασίες των τοπολογιών, καθότι μέσα στη διδακτική ύλη τις αναφέρουμε και με την Αγγλική ονομασία. Επίσης, στο τεστ, κάποιες τοπολογίες ήδη αναφέρονται με την Ελληνική τους ονομασία, οπότε θεωρήσαμε αναγκαίο ο χρήστης να δώσει βάση και στη διεθνή τους ονομασία.

Σε περίπτωση που ο χρήστης επιλέξει σωστά, τότε ακούγεται ο ήχος που δηλώνει τη σωστή επιλογή και το όνομα της τοπολογίας μεταφέρεται κάτω από την αντίστοιχη εικόνα.

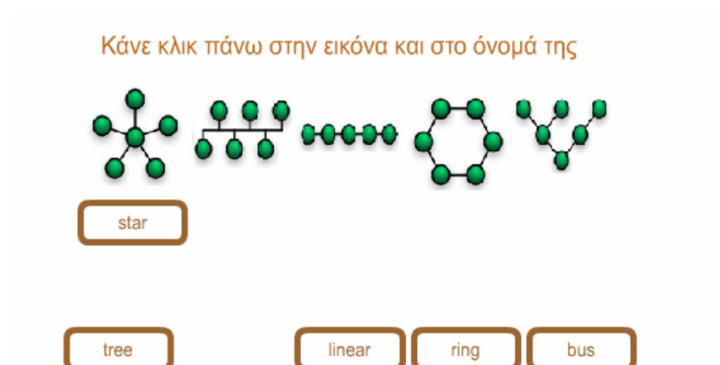
Αναλόγως, εάν ο χρήστης κάνει λάθος επιλογή, ο ήχος υποδηλώνει το λάθος και το όνομα παραμένει αμετακίνητο.



Εικόνα 6.56 Είσοδος στο Quiz

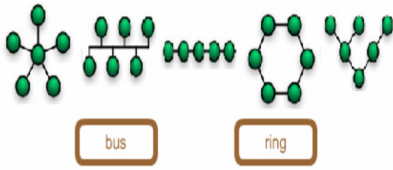


Εικόνα 6.57 Το Quiz

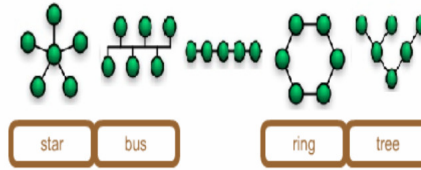


Εικόνα 6.58 Σωστή Επιλογή Ονομασίας

Κάνε κλικ πάνω στην εικόνα και στο όνομά της

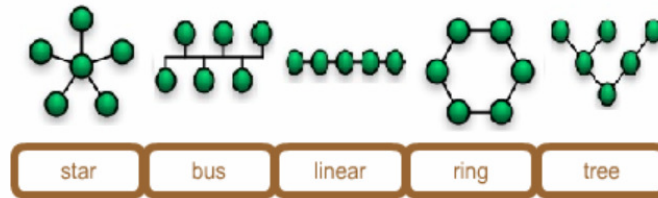


Κάνε κλικ πάνω στην εικόνα και στο όνομά της



Εικόνα 6.59 Στιγμιότυπα από το Quiz

Κάνε κλικ πάνω στην εικόνα και στο όνομά της



Συγχαρητήρια!

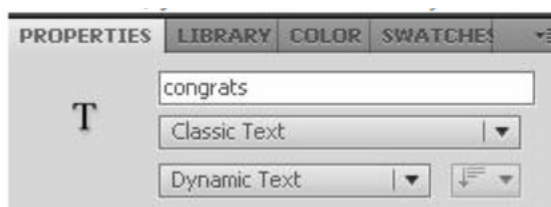
Εικόνα 6.60 Επιτυχές ολοκλήρωση του Quiz

6.6.2 Ανάπτυξη

Σε αυτήν την ενότητα θα περιγράψουμε τη διαδικασία που ακολουθήσαμε για την ανάπτυξη της εφαρμογής Quiz. Αρχικά δημιουργούμε την κυρίως σκηνή (Game.fla) η οποία περιέχει τα γραφικά του Quiz. Η κυρίως σκηνή αποτελείται από 2 frames. Το πρώτο frame είναι αυτό που αποτελεί την είσοδο του χρήστη στο quiz, καθώς το δεύτερο frame είναι το κυρίως μέρος της εφαρμογής. Ο κώδικας της εφαρμογής θα τοποθετηθεί στο αρχείο Game.as το οποίο θα περιέχει και την κυρίως class του quiz.

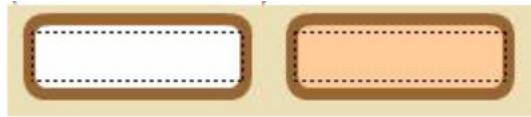
Το frame 1, Εικόνα 6.56, αποτελείται από ένα στατικό κείμενο το οποίο αποτελεί τον τίτλο της εφαρμογής, καθώς και το κουμπί play. Εάν ο χρήστης επιλέξει το κουμπί play τότε μεταφέρεται στο frame 2 της εφαρμογής. Ο τρόπος ανάπτυξης του κουμπιού play είναι ο ίδιος με αυτόν που περιγράψαμε στις προηγούμενες ενότητες. Για τον λόγο αυτόν προς αποφυγή επανάληψης δεν θα αναφέρουμε τον τρόπο ανάπτυξής του ξανά.

Στο frame 2 δημιουργούμε ένα dynamic text, το οποίο ονομάζουμε congrats. Το κείμενο αυτό θα εμφανίζεται όποτε ο χρήστης επιλέξει σωστά όλα τα ονόματα των τοπολογιών.



Εικόνα 6.61 Dynamic text

Στη συνέχεια φορτώνουμε τις εικόνες των τοπολογιών. Κάθε εικόνα είναι ονομασμένη με την ίδια ονομασία που εμφανίζεται στην εφαρμογή. Για παράδειγμα, για την τοπολογία του αστέρα/star ζητείται από το χρήστη να ταιριάζει την εικόνα με το όνομα star, καθώς η εικόνα έχει το όνομα star.jpeg. Έπειτα φτιάχνουμε τα κουτάκια μέσα στα οποία θα εμφανίζονται οι ονομασίες των τοπολογιών. Αυτό το επιτυγχάνουμε δημιουργώντας χρωματιστά κουτάκια με τα κλασικά εργαλεία ζωγραφικής, και έπειτα τοποθετώντας dynamic text μέσα στα κουτάκια αυτά. Δημιουργούμε δύο κουτάκια, ένα του οποίου το περιεχόμενο είναι λευκό και ένα του οποίου το περιεχόμενο είναι πορτοκαλί. Το λευκό είναι το αρχικό κουτάκι με την ονομασία που εμφανίζονται στην εφαρμογή, καθώς το πορτοκαλί αποτελεί την ονομασία η οποία έχει επιλεγεί από τον χρήστη, Εικόνα 6.62. Τέλος τα μετατρέπουμε σε buttons, των οποίων τα ονόματα αποτελούνται από dynamic texts και θα δηλώνονται δυναμικά μέσα στον κώδικα.



Εικόνα 6.62 Buttons

Αφού δημιουργήσαμε τα γραφικά τα οποία χρειαζόμαστε για την εφαρμογή, επόμενο βήμα είναι να δημιουργήσουμε την κυρίως κλάση της εφαρμογής. Πρώτο βήμα είναι να δηλώσουμε τις μεταβλητές τις οποίες θα χρησιμοποιήσουμε:

```
public var topology_list:Array;  
  
public var display_list:Array;  
  
public var topology_objects:Array;  
  
public var buttons:Array;  
  
public var num_items:int;  
  
public var i:int;var ii:int;  
  
public var temp1:String;var temp2:int;var rnd1:int;var rnd2:int;  
  
public var namebox:Namebox;  
  
public var answer1:String;  
  
public var answer2:String;  
  
public var num_correct:int;  
  
public var openpix:Openpix;  
  
public var frame:Frame;  
  
public var sound1:Sound1= new Sound1;  
  
public var sound2:Sound2= new Sound2;
```

Έχουμε δηλώσει 4 arrays στα οποία θα αποθηκεύσουμε τα ονόματα των εικόνων των τοπολογιών καθώς και των κουμπιών που θα φέρουν τις ονομασίες των τοπολογιών. Χρειαζόμαστε μία μεταβλητή στην οποία θα αποθηκεύσουμε τον αριθμό των αντικειμένων

της σκηνής. Όταν χρησιμοποιούμε μία for επανάληψη, συνήθως χρησιμοποιούμε την μεταβλητή i και εφόσον χρειαζόμαστε να χρησιμοποιήσουμε μία εμφωλευμένη επανάληψη, θα χρειαστούμε και την μεταβλητή ii. Οι μεταβλητές temp και rnd χρησιμοποιούνται για τυχαίες τιμές. Επίσης, στη συνέχεια θα παρουσιάσουμε και τη δημιουργία της class Openpix την οποία θα τη χρησιμοποιήσουμε για να εισάγουμε στην εφαρμογή τις εικόνες μας. Οι μεταβλητές Sound1 και Sound2 είναι για τους ήχους που αναπαράγονται κατά τη σωστή ή τη λανθασμένη επιλογή ονομασίας.

Στη συνέχεια ορίζουμε τη συνάρτηση startScreen() η οποία καλείται προκειμένου να αρχίσει το quiz. Η συνάρτηση αυτή καλείται αυτόματα με την έναρξη του quiz, και μέσα σε αυτή ορίζουμε τη λειτουργία του κουμπιού play, το οποίο θα μεταφέρει τη flash εφαρμογή στο επόμενο frame, και στη συνέχεια θα κληθεί η συνάρτηση populate().

```
public function startscreen(){
gotoAndStop(1);

bplay.addEventListener(MouseEvent.CLICK,Nav1);

function Nav1(eventObject:MouseEvent){gotoAndStop(2); populate();}

}
```

Η συνάρτηση populate() είναι υπεύθυνη για τον εντοπισμό των σωστών απαντήσεων προκειμένου να γνωρίζουμε σε πιο σημείο το quiz έχει τερματιστεί¹³. Επίσης, όλες οι εικόνες και τα κουμπιά που έχουν επιλεγεί παρακολουθούνται. Χρειαζόμαστε μία ξεχωριστή array στην οποία θα αποθηκεύσουμε τον τίτλο που θα φέρει κάθε κουμπί. Έπειτα χρειαζόμαστε μία ξεχωριστή array στην οποία θα αποθηκεύσουμε τα κουμπιά.

¹³ Το quiz τερματίζει μόνο όταν ο χρήστης έχει βρει επιτυχώς όλα τα ονόματα τοπολογιών.

```
num_correct=0;
answer1="";answer2="";
topology_list=["star","bus","linear","ring","tree"];
display_list=new Array;buttons=new Array;
```

Έπειτα, χρησιμοποιώντας μια for-επανάληψη αναθέτουμε τις τιμές που συμπεριλαμβάνει το array topology_list στο display_list. Ο λόγος που ακολουθούμε αυτήν την τακτική είναι διότι δεν μπορούμε να δηλώσουμε απλά display_list=topology_list επειδή όταν αλλάζουμε τις τιμές στη μία λίστα τότε επηρεάζονται και οι τιμές στη άλλη λίστα. Επίσης θα μπορούσαμε να δηλώσουμε απλά display_list=["star","bus","linear","ring","tree"] το οποίο θα ήταν ευκολότερο, όμως επιθυμούμε ο κώδικάς μας να είναι εύκολα επαναχρησιμοποιήσιμος, κάτι το οποίο δεν επιτυγχάνεται με αυτή την τακτική δήλωσης λίστας.

```
for(i=0;i<topology_list.length;i++){display_list[i]=topology_list[i];}
trace(topology_list);trace(display_list);
```

Επόμενο βήμα είναι να επιλέξουμε και να προβάλλουμε τα κουμπιά μας (με τα ονόματα των τοπολογιών) σε τυχαία σειρά. Την διαδικασία αυτή την εκτελούμε μέσα στη συνάρτηση scramble(), στην οποία κάνοντας χρήση της Math.random() αναπαράγουμε τυχαίους αριθμούς, οι οποίοι αποτελούν του δείκτες των arrays.

```
public function scramble(a){
for(i=0;i<30;i++){
rnd1=Math.random()*a.length;rnd2=Math.random()*a.length;
temp1=a[rnd1];a[rnd1]=a[rnd2];a[rnd2]=temp1;
```


αποτελείται από τη συνάρτηση `getpix(a,x,y)` η οποία παίρνει σαν είσοδο την εικόνα `jpeg` και τις συντεταγμένες `x-y` στις οποίες θα τοποθετηθούν οι εικόνες.

```
public function getimages(){
    topology_objects=new Array;
    for(i=0;i<topology_list.length;i++){
        openpix=new Openpix;addChild(openpix);openpix.getpix(topology_list[i],70+(i*110),120);
        topology_objects[i]=openpix;
    }
    addListeners();
}

public function getpix(a,mcx,mcy){
    pic_label=a;
    var imageLoader:Loader = new Loader();
    var image:URLRequest = new URLRequest(a+".jpg");
    imageLoader.load(image);
    addChild (imageLoader);
    this.x=mcx;
    this.y=mcy;
}
```

Επόμενος στόχος είναι να δημιουργήσουμε `listeners` προκειμένου να είμαστε ενήμεροι για τις επιλογές του χρήστη ως προς τις εικόνες και τα κουμπιά με τις ονομασίες. Για τον λόγο αυτόν, ορίζουμε τη συνάρτηση `addListeners()`

```
public function addListeners(){
    for(i=0;i<topology_objects.length;i++){
        topology_objects[i].mouseChildren = false;
```

```

topology_objects[i].addEventListener(MouseEvent.CLICK, pic_click);
}
for(i=0;i<display_list.length;i++){
buttons[i].addEventListener(MouseEvent.CLICK, but_click);
}
}

```

Όπως βλέπουμε στον κώδικα που παραθέσαμε πιο πάνω, πραγματοποιούμε μία for-επανάληψη και προσθέτουμε τους listeners pic_click στις εικόνες και but_click στα κουμπιά.

Τα κλικ των εικόνων ελέγχονται με τη συνάρτηση pic_click():

```

public function pic_click(event:MouseEvent){
frame.visible=true; frame.x=event.target.x;frame.y=event.target.y;
answer1=event.target.pic_label;
if(answer1==answer2){correct();}else{notcorrect();}
}

```

Ενώ τα κλικ των κουμπιών ελέγχονται από τη συνάρτηση but_clic():

```

public function but_click(event:MouseEvent){
for(i=0;i<buttons.length;i++){buttons[i].gotoAndStop(1);}
event.target.gotoAndStop(2);
answer2=event.target.topologyname.text;
if(answer1==answer2){correct();}else{notcorrect();}
}

```

Αφού στήσαμε το σκηνικό μας, απαραίτητο είναι να δημιουργήσουμε μία συνάρτηση με την οποία θα ελέγχουμε τις σωστές απαντήσεις του χρήστη. Για τον λόγο αυτό, ορίζουμε τη

συνάρτηση correct() στην οποία όχι μόνο αναγνωρίζονται οι σωστές και οι λάθος απαντήσεις, αλλά και αναπαράγεται ο αντίστοιχος ήχος (Sound1, Sound2):

```
public function correct(){num_correct++;sound1.play();

for(i=0;i<buttons.length;i++){if(buttons[i].topologyname.text==answer2){break;}}

for(ii=0;ii<topology_list.length;ii++){if(topology_list[ii]==answer2){break;}}

buttons[i].x=topology_objects[ii].x-5;

buttons[i].y=topology_objects[ii].y+110;

buttons[i].gotoAndStop(1);

buttons[i].removeEventListener(MouseEvent.CLICK, but_click);

topology_objects[ii].removeEventListener(MouseEvent.CLICK, pic_click);

answer1="";answer2="";

if(num_correct==topology_objects.length){congrats.text="Συγχαρητήρια!";}

}
```

Όπως φαίνεται στην παραπάνω κώδικα, χρησιμοποιούμε την for-επανάληψη για να βεβαιώσουμε ότι το buttons[i] αναφέρεται στο κουμπί το οποίο έκανε κλικ ο χρήστης. Την ίδια τακτική ακολουθούμε και για να διαβεβαιώσουμε ότι το topology_list[ii] και το topology_objects[ii] αναφέρεται στην εικόνα την οποία επιλέχτηκε από το χρήστη. Στη συνέχεια μετακινούμε το κουμπί και το τοποθετούμε κάτω από την αντίστοιχη εικόνα. Τα -5 και +110 τα χρησιμοποιούμε για σκοπούς ευθυγράμμισης. Χρησιμοποιούμε τη συνάρτηση gotoAndStop() προκειμένου να επισημάνουμε το κουμπί.¹⁴ Αφαιρούμε το listener από την εικόνα έτσι ώστε να μην θεωρείται πλέον επιλεγμένη. Στη συνέχεια μηδενίζουμε τις τιμές των answer1 και answer2, για να τις επαναχρησιμοποιήσουμε στις επόμενες απαντήσεις του χρήστη. Χρησιμοποιούμε το if για να ελέγξουμε την περίπτωση που έχουν βρεθεί όλες οι

¹⁴ Σημείωση: σε προηγούμενο στάδιο δημιουργήσαμε δύο διαφορετικά κουμπιά, ένα με λευκό φόντο και ένα με πορτοκαλί. Το κουμπί με το πορτοκαλί φόντο χρησιμοποιείται για να επισημάνουμε την επιλογή του χρήστη.

σωστές απαντήσεις, οπότε να προβάλλουμε τη λεζάντα με τα «Συγχαρητήρια» προς το χρήστη.

Προκειμένου να επεξεργαστούμε τις λανθασμένες απαντήσεις, δηλώνουμε τη συνάρτηση `notcorrect()`, στην οποία βεβαιώνουμε ότι έχει επιδεχθεί από το χρήστη και εικόνα και κουμπί, αναπαράγουμε τον ήχο που υποδηλώνει λάθος επιλογή, και στη συνέχεια μηδενίζουμε τις `answer1`, `answer2` για να τις επαναχρησιμοποιήσουμε στη συνέχεια. Ο κώδικας είναι:

```
public function notcorrect(){
    if(answer1!="" && answer2!=""){
        sound2.play();
        answer1="";answer2="";
        frame.visible=false;
        for(i=0;i<display_list.length;i++){buttons[i].gotoAndStop(1);}//fori
    }
}
```

6.7 Βασικός Σκελετός Εφαρμογής

Η τακτική που ακολουθήσαμε για την ανάπτυξη του εκπαιδευτικού λογισμικού για το μάθημα των Δικτύων στηρίζεται στην λογική της ανάπτυξης ενός σκελετού για το λογισμικό, και πάνω σε αυτό η πρόσθεση περαιτέρω εφαρμογών. Ασφαλώς, η ευκολότερη τακτική θα ήταν να υλοποιήσουμε ολόκληρο το λογισμικό σαν ένα μοναδικό κομμάτι, υπό την έννοια ότι οι υποεφαρμογές που αναφέραμε στις προηγούμενες ενότητες δεν θα αποτελούσαν αυτόνομα κομμάτια, αλλά ένα ενιαίο κομμάτι flash εφαρμογής. Παρόλα αυτά, επιλέξαμε να τα υλοποιήσουμε ξεχωριστά διότι η συντήρηση/αλλαγή του κώδικα και ειδικότερα των υποεφαρμογών είναι ευκολότερη όταν οι εφαρμογές αυτές αποτελούνται από ξεχωριστές οντότητες κώδικα.

Στις προηγούμενες ενότητες παρουσιάσαμε τις υποεφαρμογές που αναπτύξαμε καθώς και τη διαδικασία ανάπτυξής τους. Σε αυτήν την ενότητα θα περιγράψουμε το βασικό σκελετό πάνω στον οποίο προστέθηκαν και συνδυάστηκαν οι υποεφαρμογές.

Αρχικό βήμα για τη δημιουργία του σκελετού είναι η ανάπτυξη της βασικής σκηνής πάνω στην οποία θα προστεθούν οι υποεφαρμογές. Ο σκελετός αυτός αποτελείται από 2 κομμάτια: μία απλή σκηνή η οποία περιέχει τα βασικά κουμπιά «Βοήθεια», «Μουσική», «Πίσω», «Skip Intro» (μόνο για το Intro) και τη σκηνή η οποία περιέχει τις ενότητες.

6.8.1 Απλή σκηνή

Για την ανάπτυξη της βασικής σκηνής αρχικά δημιουργήσαμε το φόντο το οποίο θα παρουσιάζεται κατά τη διάρκεια της εκτέλεσης της εφαρμογής, Εικόνα 6.63. Ανάλογα το σημείο εκτέλεσης της εφαρμογής, φορτώνουμε στη σκηνή την αντίστοιχη υποεφαρμογή. Για παράδειγμα, κατά την εισαγωγή στο λογισμικό, φορτώνουμε στην κυρίως σκηνή την υποεφαρμογή intro:

```
fl_Loader = new Loader();  
fl_Loader.load(new URLRequest("intro.swf"));  
fl_Loader.x=Xpos;  
fl_Loader.y=Ypos;  
addChild(fl_Loader);  
introMusic.load(introSong);
```



Εικόνα 6.63 Βασική Σκηγή

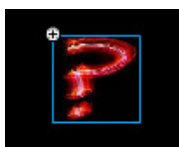
Επόμενο βήμα είναι η δημιουργία των κουμπιών που θα εμφανίζονται κατά τη διάρκεια εκτέλεσης της εφαρμογής, ανεξαρτήτως υποεφαρμογής ή ενότητας. Τα κουμπιά αυτά είναι «Βοήθεια», «Μουσική», «Πίσω»

6.8.2 Κουμπιά

Όλα τα κουμπιά που δημιουργήσαμε για το εκπαιδευτικό λογισμικό των Δικτύων, έχουν αναπτυχθεί με την ίδια τακτική. Προς αποφυγήν επαναλήψεως, θα περιγράψουμε την όλη διαδικασία μόνο για ένα από όλα τα κουμπιά. Η μόνη διαφορά σε κάθε κουμπί είναι το γραφικό της εικόνας που χρησιμοποιήσαμε, καθώς και ο κώδικας που εκτελείται από το κάθε κουμπί. Στο παρακάτω κείμενο περιγράψουμε τη διαδικασία ανάπτυξης για το κουμπί «Βοήθεια».

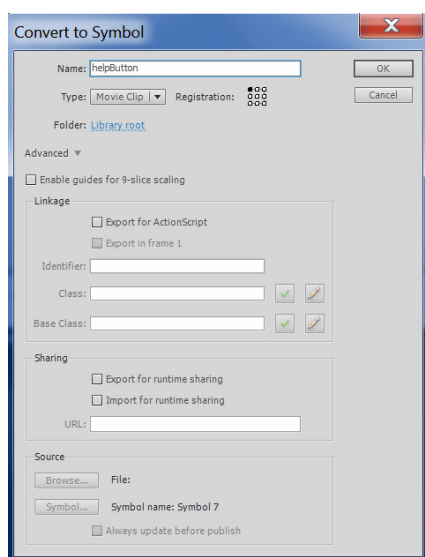
Σχεδιασμός Κουμπιών

Το πρώτο βήμα είναι ο σχεδιασμός των γραφικών για το κουμπί. Αρχικά δημιουργούμε μία εικόνα, η οποία αποτελείται από ένα Αγγλικό ερωτηματικό, Εικόνα 6.64



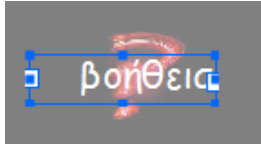
Εικόνα 6.64 Αγγλικό Ερωτηματικό

Έπειτα, μετατρέπουμε την εικόνα και σε movie clip, Εικόνα 6.65, και σε button. Προσοχή, μετατρέπουμε την εικόνα και σε movie clip καθότι θέλουμε να δημιουργήσουμε οπτικά εφέ, τα οποία δεν είναι εφικτό να τα πραγματοποιήσουμε κατευθείαν σε ένα button.



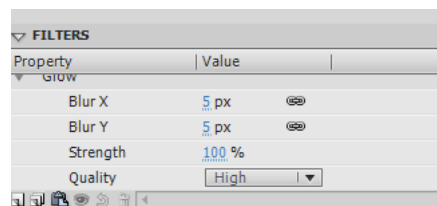
Εικόνα 6.65 Δημιουργία Κουμπιού

Εφόσον μετατρέψαμε την εικόνα σε movie clip, το επόμενο βήμα είναι να δημιουργήσουμε τα οπτικά εφέ. Αρχικά ανοίγουμε το movie clip helpButton, και αρχίζουμε να επεξεργαζόμαστε τα frames του. Δημιουργούμε 4 layers, καθένα από τα οποία θα περιέχει διαφορετικές ιδιότητες και γραφικά. Αρχικά στο layer 1, δημιουργούμε πάνω στο κουμπί το κείμενο «βοήθεια», Εικόνα 6.66.



Εικόνα 6.66. Κείμενο

Στη συνέχεια επιλέγουμε το κείμενο και το μετατρέπουμε σε movie clip. Εφόσον το κείμενο είναι πλέον movie clip, εκμεταλλευόμαστε τις δυνατότητες που προσφέρει το flash, προκειμένου να δημιουργήσουμε ένα οπτικό εφέ. Σε πιο λεπτομέρεια, δημιουργούμε ένα keyframe στο frame 8, και επιλέγουμε στα properties του movie clip «βοήθεια» τα φίλτρα glow και blur (Εικόνα 6.67) .



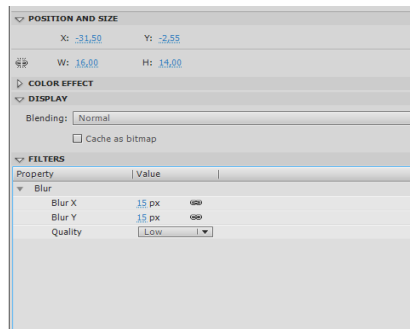
Εικόνα 6.67 Blur

Έπειτα δημιουργούμε άλλο ένα keyframe στο frame 28. Στα frames πριν το 8 και μετά το 28, μηδενίζουμε τις τιμές του blur και του glow. Αυτό έχει ως αποτέλεσμα, Η λέξη «βοήθεια» να εμφανίζεται μόνο στα frames 8-28.

Στο layer 2, δημιουργούμε ένα κόκκινο τετράγωνο (με τα κλασικά εργαλεία ζωγραφικής), το οποίο το μετατρέπουμε σε movie clip. Στη συνέχεια το τοποθετούμε στην πάνω αριστερή γωνία του κουμπιού, και δίνουμε στη μεταβλητή blur την τιμή 15, Εικόνες 6.68 - 6.69.

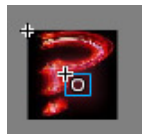


Εικόνα 6.68



Εικόνες 6.69

Έπειτα δημιουργούμε ένα keyframe στο frame 15, και μετακινούμε το τετράγωνό μας στο κέντρο του κουμπιού, Εικόνα 6.70



Εικόνα 6.70

Σε αυτό το σημείο θα αλλάξουμε ξανά τις τιμές του blur, τις οποίες θα τις αυξήσουμε στο 25. Στη συνέχεια, δημιουργούμε άλλο ένα keyframe στο frame 28, μετακινούμε το τετράγωνο στην κάτω δεξιά γωνία και επαναφέρουμε τις τιμές του blur στο 15, Εικόνα 6.71.



Εικόνα 6.71

Το αποτέλεσμα αυτής της τακτικής είναι η μετακίνηση του κόκκινου τετραγώνου από την πάνω αριστερή γωνία (αρχικό frame), στη μέση (frame 15) στην οποία το χρώμα του γίνεται

πιο αγνό, και στο τέλος στην κάτω αριστερή θέση (frame 28) στην οποία το χρώμα του επαναφέρεται. Προκειμένου να ομαλοποιήσουμε την κίνηση αυτή, επιλέγουμε τα frames 0-28 και κάνοντας δεξί κλικ επιλέγουμε create classic tween, Εικόνα 6.72.



Εικόνα 6.72

Τη διαδικασία αυτή την επαναλαμβάνουμε και στο layer 3, στο οποίο δημιουργούμε ένα κόκκινο τετράγωνο με αρχική θέση την κάτω δεξιά γωνία, το οποίο έχει τελική θέση πάνω αριστερά. Τέλος στο layer 4, δημιουργούμε διάφορα keyframes στα οποία αλλάζουμε τις τιμές των blur για τα movie clips που έχουμε στη σκηνή.

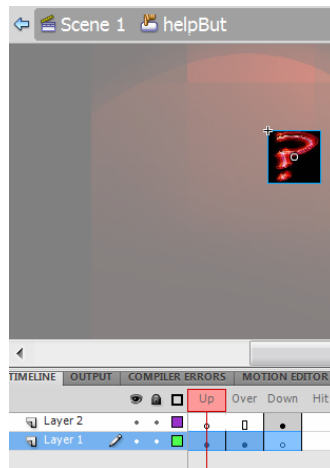
Το αποτέλεσμα αυτής της διαδικασίας είναι ένα movie clip, το οποίο αποτελείται από το βασικό μας φόντο (ερωτηματικό), από τη λεζάντα (βοήθεια) η οποία αναβοσβήνει, και από φωτεινά τετράγωνα τα οποία περιστρέφονται γύρω του. Στην Εικόνα 6.73 προβάλλουμε μερικά στιγμιότυπα από το movie clip.



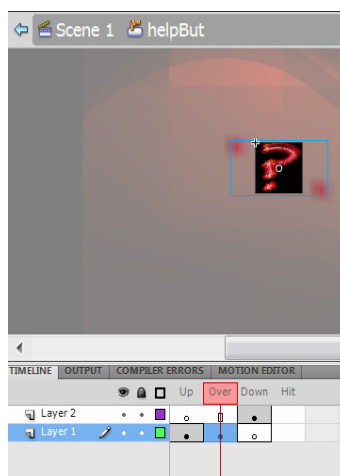
Εικόνα 6.73

Μετά τη δημιουργία του movie clip, επόμενο βήμα η ολοκλήρωση του κουμπιού «βοήθεια». Κάνοντας διπλό κλικ στο button που δημιουργήσαμε προηγουμένως, εμφανίζονται τα frames του button (up, over, down), κάθε ένα από τα οποία αντιστοιχεί στην αντίστοιχη κίνηση που πραγματοποιεί ο χρήστης. Στόχος μας είναι να εμφανίζεται το movie clip που δημιουργήσαμε, κατά τη διάρκεια που το ποντίκι του χρήστη περνάει πάνω από το κουμπί.

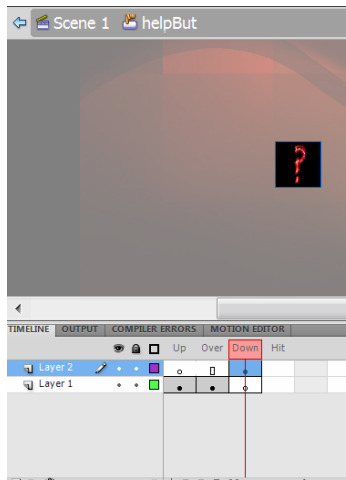
Για τον λόγο αυτό στο frame over, προσθέτουμε στη σκηνή το movie clip. Στο frame up, μικραίνουμε ελαφρώς το ερωτηματικό της εικόνας, για να δώσουμε την ψευδαίσθηση της κίνησης, Εικόνες 6.74 – 6.76.



Εικόνα 6.74 Up frame



Εικόνα 6.75 Over frame

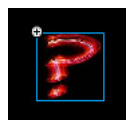


Εικόνα 6.76 Down frame

Λειτουργικότητα και Προγραμματισμός Κουμπιών

Όπως προαναφέραμε, τα κουμπιά που δημιουργήσαμε για το σκελετό του εκπαιδευτικού λογισμικού, είναι το κουμπί «βοήθεια», «μουσική», «πίσω» και «skip intro» κάθε ένα από τα οποία εκτελεί διαφορετικές λειτουργίες.

- Κουμπί Βοήθεια



Εικόνα 6.77

Το κουμπί της βοήθειας, Εικόνα 6.77, σχεδιάστηκε με σκοπό να προσφέρει βασικές πληροφορίες στο χρήστη ως προς το περιεχόμενο της σελίδας της εφαρμογής στην οποία βρίσκεται εκείνη τη στιγμή, καθώς και τις επιλογές που έχει ο χρήστης. Η λογική η οποία ακολουθήσαμε, για το κουμπί της βοήθειας, είναι πατώντας το ο χρήστης, να εμφανίζεται στην οθόνη ένα κείμενο το οποίο περιγράφει τις αντίστοιχες πληροφορίες. Όταν ο χρήστης το επιλέξει δεύτερη φορά, τότε το κείμενο εξαφανίζεται από την οθόνη. Με βάση αυτά τα χαρακτηριστικά, το πρώτο βήμα είναι να ορίσουμε έναν listener, ο οποίος θα ελέγχει εάν το

κουμπί αυτό επιλέχθηκε από τον χρήστη. Εάν το κουμπί έχει επιλεγθεί, τότε καλείται η `fl2_ClickToPosition()` συνάρτηση.

```
helpButton.addEventListener(MouseEvent.CLICK, fl2_ClickToPosition);
```

Στη συνέχεια ορίζουμε την μεταβλητή `fl2_TF` στην οποία θα αποθηκεύσουμε το κείμενο που θα εμφανίζεται στην οθόνη.

```
var fl2_TF:TextField;
```

```
var fl2_TextToDisplay:String = "Το πρόγραμμα αυτό απευθύνεται σε χρήστες οικείους και μη με το γνωστικό αντικείμενο των Δικτύων. Παρακαλώ επιλέξτε το επίπεδό σας, λαμβάνοντας υπόψιν ότι ως αρχάριος το εκπαιδευτικό υλικό στηρίζεται κυρίως σε εισαγωγικές έννοιες, ενώ ως αρχάριος σε πιο προχωρημένα θέματα.
```

```
Κατα τη διάρκεια πλοήγησή σας μπορείτε να ακούσετε μουσική, κάνοντας κλικ στο κουμπί ΜΟΥΣΙΚΗ.
```

```
Μπορείτε να επιστρέψετε σε κάποια προηγούμενη ενότητα επιλέγοντας το κουμπί ΠΙΣΩ"
```

```
var fl2_ToLoad: Boolean = true;
```

Έπειτα, ορίζουμε τη `fl2_ClickToPosition()` συνάρτηση, η οποία είναι υπεύθυνη για την προβολή του κειμένου στην οθόνη. Μέσα σε αυτή τη συνάρτηση ορίζουμε τις συντεταγμένες στις οποίες θα εμφανιστεί το κείμενο, καθώς και τη γραμματοσειρά και το φόντο του κειμένου.

```
function fl2_ClickToPosition(event:MouseEvent):void
```

```
{if (fl2_ToLoad){
```

```
fl2_TF = new TextField();
```

```
fl2_TF.autoSize = TextFieldAutoSize.LEFT;
```

```
fl2_TF.background = true;
```

```
fl2_TF.backgroundColor= 0x000000;
```

```
fl2_TF.border = true;

fl2_TF.wordWrap = true;

// fl2_TF.fontStyle = BOLD;

fl2_TF.textColor = 0xFF0000;

fl2_TF.width = 650;

fl2_TF.height = 140;

fl2_TF.x = 30;

fl2_TF.y = 450;

fl2_TF.text = fl2_TextToDisplay;

addChild(fl2_TF);

fl2_ToLoad = false;

}

else

{

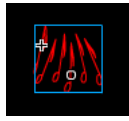
fl2_ToLoad = true

removeChild(fl2_TF);

}}
```

Όπως παρατηρούμε στον κώδικα, έχουμε χρησιμοποιήσει ένα if-statement, το οποίο ελέγχει εάν το κουμπί πατιέται για πρώτη φορά ή όχι. Στην ουσία πριν πατηθεί το κουμπί για πρώτη φορά ο loader είναι true, το οποίο δηλώνει ότι το κείμενο πρέπει να φορτωθεί και να προβληθεί στην οθόνη. Αφού πατηθεί για πρώτη φορά το κείμενο, ο loader γίνεται false, το οποίο δηλώνει ότι το κείμενο πρέπει να αφαιρεθεί από τον loader, προκειμένου να μην είναι πλέον εμφανής.

- Κουμπί Μουσική



Εικόνα 6.78

Σκοπός για το κουμπί της μουσικής, Εικόνα 6.78, είναι όταν ο χρήστης το επιλέγει, να αρχίζει η αναπαραγωγή μουσικής. Εφόσον έχει επιλεγεί ήδη μια φορά και ξανά επιλεγεί, η αναπαραγωγή σταματά. Για να επιτύχουμε αυτή τη λειτουργία, αρχικά πρέπει να δηλώσουμε έναν listener για το κουμπί αυτό, έτσι ώστε εάν γίνει κλικ στο κουμπί καλείται η συνάρτηση `fl_ClickToPlayStopSound()`.

```
musicButton.addEventListener(MouseEvent.CLICK, fl_ClickToPlayStopSound);
```

Στη συνέχεια δημιουργούμε τη συνάρτηση `fl_ClickToPlayStopSound()` η οποία είναι η υπεύθυνη για την φόρτωση και την αναπαραγωγή της μουσικής.

```
var fl_SC:SoundChannel;

var fl_ToPlay:Boolean = true;

function fl_ClickToPlayStopSound(evt:MouseEvent):void

{

if(fl_ToPlay)

{

var s:Sound = new Sound(new URLRequest("music/mousiki.mp3"));

fl_SC = s.play();

}

else
```

```
{  
fl_SC.stop();  
}  
fl_ToPlay = !fl_ToPlay;  
}
```

Αρχικά ορίζουμε μια boolean μεταβλητή fl_ToPlay, η οποία έχει αρχική τιμή true. Όταν ο χρήστης κάνει κλικ στο κουμπί της μουσικής, και καλείται η συνάρτηση fl_ClickToPlayStopSound() με την fl_ToPlay==true. Η μεταβλητή αυτή όταν είναι true, δηλώνει ότι πρέπει να φορτωθεί στον loader η μουσική, και στη συνέχεια να αναπαραχθεί. Με το που εκτελεστούν αυτές οι ενέργειες αναθέτουμε στην fl_ToPlay την τιμή false, έτσι ώστε την επόμενη φορά που θα κάνει κλικ ο χρήστης στο κουμπί, να αφαιρεθεί από τον loader η μουσική, και να σταματήσει η αναπαραγωγή της. Για τον λόγο αυτό κάνουμε χρήση των if-statements τα οποία ελέγχουν την τιμή της fl_ToPlay.

- Κουμπί Πίσω



Εικόνα 6.79

Η λειτουργικότητα για το κουμπί πίσω Εικόνα 6.79, είναι ο χρήστης να μπορεί να πλοηγείται στις προηγούμενες σελίδες της εφαρμογής. Δηλαδή, κάνοντας κλικ στο κουμπί αυτό, ο χρήστης μεταφέρεται στην αμέσως προηγούμενη ενότητα της εκπαιδευτικής εφαρμογής. Προκειμένου να επιτύχουμε αυτή την λειτουργία αναπτύξαμε των παρακάτω κώδικα:


```

backButton.addEventListener(MouseEvent.CLICK, f3_ClickToLoadUnloadSWF);

var f3_ToLoad:Boolean = true;

var f3_Loader:Loader;

function f3_ClickToLoadUnloadSWF(event:MouseEvent):void
{
    if(f3_ToLoad)
    {
        f3_Loader = new Loader();
        f3_Loader.load(new URLRequest("intro.swf"));
        f3_Loader.x=Xpos;
        f3_Loader.y=Ypos;
        addChild(f3_Loader);
    }
    else
    {
        f3_Loader.unload();
        removeChild(f3_Loader);
        f3_Loader = null;
    }
    f3_ToLoad = !f3_ToLoad;
}

```

Κάνοντας κλικ στο κουμπί «πίσω» ο χρήστης, καλείται η συνάρτηση `f3_ClickToLoadUnloadSWF` (εφόσον έχουμε ορίσει έναν listener για αυτό το κουμπί). Στη συνάρτηση αυτή υλοποιούμε την φόρτωση της κατάλληλης flash υποεφαρμογής, η οποία αντιστοιχεί στην προηγούμενη ενότητα του λογισμικού. Στον κώδικα που παραθέτουμε φορτώνεται το `intro` του εκπαιδευτικού λογισμικού. Το flash αρχείο που θα φορτωθεί είναι διαφορετικό, και εξαρτάται από το σημείο του λογισμικού στο οποίο θα κάνει κλικ ο

χρήστης. Η διαδικασία που ακολουθούμε με τους loaders και τις Boolean μεταβλητές στα if-statements είναι η ίδια με την οποία περιγράψαμε για τα προηγούμενα κουμπιά.

- Κουμπί Skip Intro

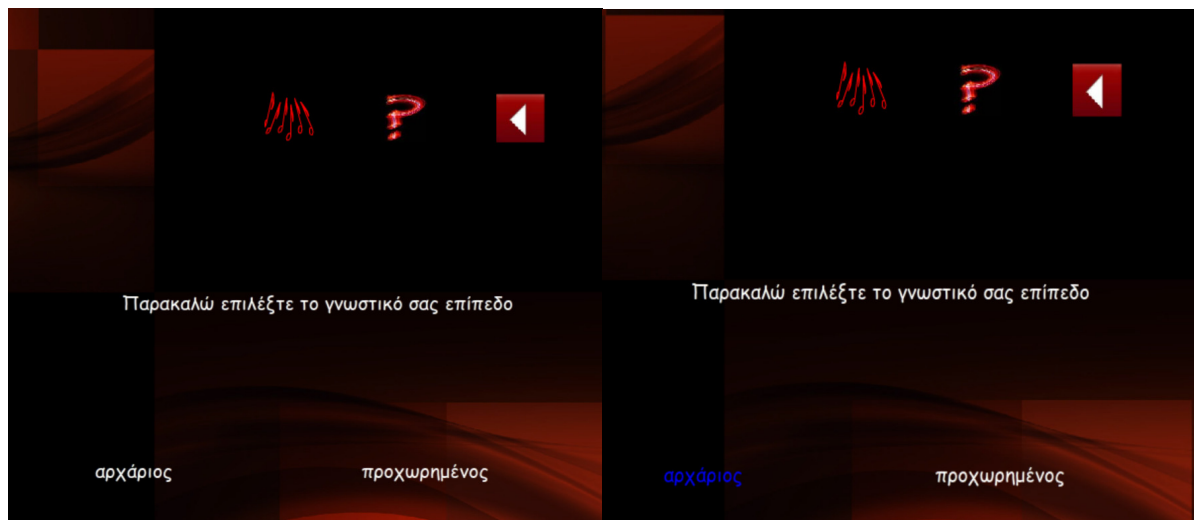


Εικόνα 6.80

Το κουμπί Skip Intro Εικόνα 6.80, σε σχέση με τα υπόλοιπα κουμπιά, εμφανίζεται μόνο στο intro του λογισμικού. Ο λόγος ύπαρξης του κουμπιού αυτού είναι ο χρήστης να μπορεί να επιλέγει το κλείσιμο της intro ταινίας και να προχωράει κατευθείαν στο κυρίως κομμάτι του λογισμικού. Ο κώδικας του κουμπιού είναι παρεμφερής με τον κώδικα του κουμπιού «πίσω». Για τον λόγο αυτό, και προς αποφυγή επαναλήψεως των παραθέτουμε στο τελευταίο κομμάτι της πτυχιακής, στο κεφάλαιο «Κώδικας».

6.8.3 Επιλογή Γνωστικού Επιπέδου

Όπως προαναφέραμε, το εκπαιδευτικό λογισμικό το οποίο παρουσιάζουμε δίνει τη δυνατότητα στο χρήστη να επιλέξει το γνωστικό του επίπεδο. Αναλόγως με την επιλογή αυτή το διδακτικό περιεχόμενο διαμορφώνεται αντίστοιχα. Για τον λόγο αυτό, κατά την έναρξη του εκπαιδευτικού λογισμικού, ο χρήστης καλείται να επιλέξει το γνωστικό του επίπεδο. Στιγμιότυπα απεικονίζονται στην Εικόνα 6.81



Εικόνα 6.81 Επιλογή γνωστικού επιπέδου

Προκειμένου να επιτύχουμε αυτή την λειτουργικότητα, δημιουργήσαμε δύο κουμπιά, «αρχάριος» και «προχωρημένος». Αναλόγως το κουμπί το οποίο θα επιλέξει ο χρήστης, θα συνεχίσει και την πλοήγησή του στο λογισμικό. Για τα κουμπιά αυτά, χρησιμοποιήσαμε την ίδια λογική προγραμματισμού: φορτώνουμε στη σκηνή την υποεφαρμογή που αντιστοιχεί στο κουμπί το οποίο επιλέχτηκε. Για παράδειγμα, ο κώδικας που χρησιμοποιήθηκε για την επιλογή «αρχάριος» είναι:

```
arxarios.addEventListener(MouseEvent.CLICK, fl_ClickToLoadUnloadSWF);
```

```
var Xpos:Number=10;
```

```
var Ypos:Number=7;
```

```
var fl_Loader:Loader;
```

```
var fl_ToLoad:Boolean = true;
```

```
function fl_ClickToLoadUnloadSWF(event:MouseEvent):void
```

```
{
```

```
if(fl_ToLoad)
```

```

{
fl_Loader = new Loader();
fl_Loader.load(new URLRequest("arxarios.swf"));
fl_Loader.x=Xpos;
fl_Loader.y=Ypos;
addChild(fl_Loader);
}
else
{
fl_Loader.unload();
removeChild(fl_Loader);
fl_Loader = null;
}
fl_ToLoad = !fl_ToLoad;
}

```

Αρχικά ορίζουμε έναν listener στο κουμπί «arxarios» το οποίο ελέγχει εάν έχει γίνει κλικ, και τότε καλείται η `fl_ClickToLoadUnloadSWF()`. Στη συνέχεια ορίζουμε τις μεταβλητές `Xpos`, `Ypos` οι οποίες αποτελούν τις συντεταγμένες στις οποίες θα φορτωθεί η flash υποεφαρμογή. Η λογική που ακολουθούμε στη συνάρτηση `fl_ClickToLoadUnloadSWF()` είναι η ίδια με αυτήν που περιγράψαμε στις προηγούμενες ενότητες, κατά την οποία πραγματοποιούμε φόρτωση κάποιας flash εφαρμογής στην κυρίως σκηνή.

6.9.1 Ενότητες Μενού

Δημιουργήσαμε ένα μενού το οποίο διαμορφώνεται από τις διδακτικές ενότητες που αντιστοιχούν στο μάθημα των Δικτύων, καθώς και από τις υποεφαρμογές που αναπτύχθηκαν

για αυτή τη διδακτική ύλη. Οι διδακτικές ενότητες αποτελούνται από την «Εισαγωγή» (αρχάριος/προχωρημένος), «Σήματα» (αρχάριος), «Μετάδοση» (αρχάριος), «Τοπολογίες» (αρχάριος), «Τεχνολογία Σημείου» (αρχάριος), «Μοντέλα Δικτύων» (προχωρημένος), «Σφάλματα» (προχωρημένος), «Πρότυπα» (προχωρημένος), «Πολυπλεξία» (προχωρημένος), «Επικοινωνία» (αρχάριος/προχωρημένος), «Quiz» (αρχάριος/προχωρημένος), «Παιχνίδι» (αρχάριος/προχωρημένος), «Τεστ» (αρχάριος/προχωρημένος), «Βίντεος» (αρχάριος/προχωρημένος), Εικόνες 6.82 – 6.83.



Εικόνα 6.82 Κυρίως Μενού Αρχάριος



Εικόνα 6.83 Κυρίως Μενού Προχωρημένος

6.9.2 Κυρίως Σκελετός

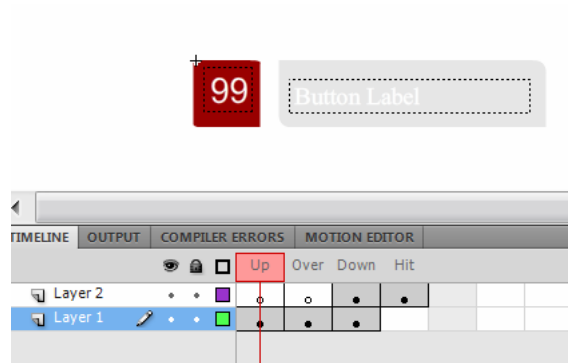
Το κυρίως μενού αποτελείται από πολλαπλά πανομοιότυπα κουμπιά, καθένα από τα οποία αναπαριστά μία ενότητα του λογισμικού. Το πρώτο βήμα για την ανάπτυξη του μενού, ήταν η δημιουργία του κυρίως κουμπιού, Εικόνα 6.84. Η λογική που ακολουθήσαμε ήταν ότι δημιουργήσαμε ένα κουμπί, το οποίο μέσω προγραμματισμού αναπαράγαμε και δώσαμε διαφορετικές ονομασίες και λειτουργίες.



Εικόνα 6.84

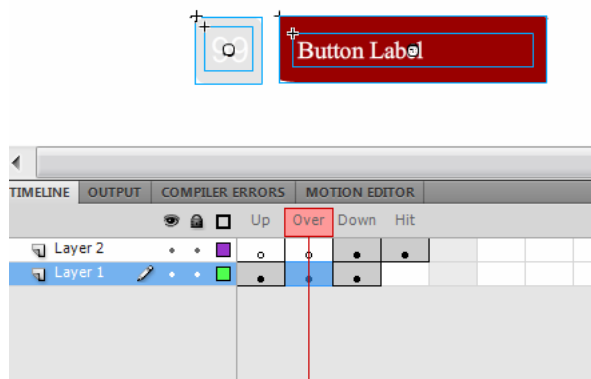
Αφού σχεδιάσαμε το κουμπί, το μετατρέψαμε σε button και αρχίσαμε την επεξεργασία των frames του. Πρώτο βήμα είναι η δημιουργία δύο dynamic text πάνω στο κουμπί: ένα για τον

αριθμό (σειρά) και ένα για τον τίτλο. Στη συνέχεια για το frame up αφήσαμε το γραφικό του κουμπιού ίδιο, Εικόνα 6.85



Εικόνα 6.85

Έπειτα, για το frame down, πραγματοποιήσαμε εναλλαγή των χρωμάτων του κουμπιού, Εικόνα 6.86



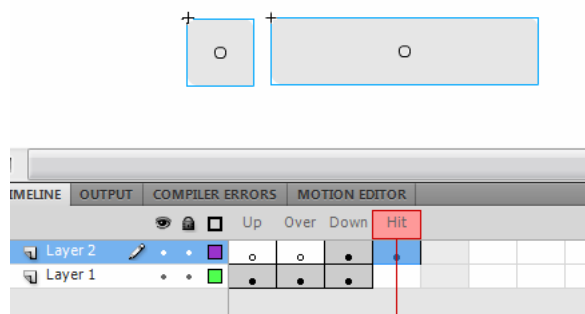
Εικόνα 6.86

Και στο frame down, επαναφέραμε τα αρχικά χρώματα (frame up) αυτή τη φορά δημιουργώντας μία μικρή σκιά, Εικόνα 6.87



Εικόνα 6.87

Τέλος, για το frame hit, ορίσαμε το πεδίο του αριθμού αλλά και του τίτλου να έχουν το ίδιο χρώμα, Εικόνα 6.88



Εικόνα 6.88

Το επόμενο βήμα είναι να δημιουργήσουμε ένα array στο οποίο θα αποθηκεύσουμε τους τίτλους των κουμπιών που θα αναπαράγουμε. Για παράδειγμα, για το επίπεδο του προχωρημένου, ορίζουμε:

```
var menu_label:Array = new Array(" Εισαγωγή ", "Μοντέλα Δικτύων ", " Σφάλματα",  
"Πρότυπα", "Πολυπλεξία", "Επικοινωνία", "Quiz",  
"Παιχνίδι", "Test", "Βίντεο");
```

Έπειτα, στόχος μας είναι να αναπαράγουμε τόσα κουμπιά όσοι οι τίτλοι που έχουμε ορίσει στο array menu_label και να τοποθετήσουμε στα κουμπιά αυτά τον αντίστοιχο τίτλο καθώς και το αντίστοιχο νούμερο. Για να το επιτύχουμε αρχικά διαχωρίζουμε τα strings που

περιέχει το array menu_label, και στη συνέχεια τα τοποθετούμε σε μία μήτρα, each_substring.

```
var each_substring:Array = menu_label[i].split(" ");  
btn.flashmo_button_label.fm_label.text = each_substring[0];
```

Αυτή η τακτική όχι μόνο μας βοηθάει να ξεχωρίσουμε τους τίτλους των κουμπιών, αλλά και να μετρήσουμε το πλήθος τους, προκειμένου στη συνέχεια να τα αναπαράγουμε χρησιμοποιώντας των δείκτη της μήτρας.

```
btn.flashmo_button_number.fm_number.text = "0" + (i + 1);  
main_menu.addChild(btn);
```

Όλη αυτή η διαδικασία, προφανώς, τοποθετείται μέσα σε μια for-επανάληψη :

```
for (i = 0; i < total; i++)  
{  
var btn = new flashmo_button();  
btn.name = "btn" + i;  
btn.x = fm_button.x;  
btn.y = fm_button.y + i * ( fm_button.height + 10 ) + 100;  
btn.alpha = 0;  
btn.buttonMode = true;  
btn.item_no = i;  
btn.flashmo_click_area.addEventListener( Event.ENTER_FRAME, btn_enter );  
  
var each_substring:Array = menu_label[i].split(" ");
```

```

btn.flashmo_button_label.fm_label.text = each_substring[0];

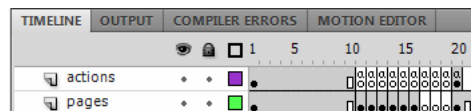
btn.item_url = each_substring[1];

btn.flashmo_button_number.fm_number.text = "0" + (i + 1);

main_menu.addChild(btn);
}

```

Στη συνέχεια, στην κυρίως σκηνή, δημιουργούμε δύο layers, ένα για τα γραφικά και ένα για τον κώδικα. Έπειτα, ορίζουμε και στα δύο layers keyframes, κάθε ένα από τα οποία αντιστοιχεί σε ένα από τα κουμπιά του μενού, Εικόνα 6.89



Εικόνα 6.89

Για τα εφέ τα οποία χρησιμοποιήσαμε κατα την έναρξη του κυρίως μενού (σταδιακή αναδίπλωση του μενού, transitions. κτλ) καθώς και για τις μπάρες πλοήγησης στο κείμενο, χρησιμοποιήσαμε έτοιμα εφέ από την εταιρία *flasmo*. Ο κώδικας για τα εφέ αυτά παρέχεται στο τελευταίο κεφάλαιο.

Διδακτικές ενότητες

Για κάθε μία από τις διδακτικές ενότητες δημιουργήσαμε το αντίστοιχο xml αρχείο μέσα στο οποίο αποθηκεύσαμε τη διδακτική ύλη. Ο λόγος που ακολουθήσαμε αυτή την τακτική είναι ότι η πραγματοποίηση οποιασδήποτε αλλαγής στη διδακτική ύλη, είναι ευκολότερο να πραγματοποιηθεί σε εξωτερικά αρχεία παρά κατευθείαν στον κώδικα. Επίσης, τα xml αρχεία επιτρέπουν την τροποποίηση και τον καθορισμό της παρουσίασης του κειμένου, μέσω css κλάσεων. Για παράδειγμα,

```
<content><![CDATA[<span class="title">Εισαγωγή στις Τοπολογίες</span>
```

```
<p align="justify">Τοπολογία δικτύου ονομάζεται η μορφή της σύνδεσης μεταξύ των κόμβων ενός δικτύου. Οι τοπολογίες είναι είτε φυσικές είτε λογικές. Τα κυριότερα είδη τοπολογιών είναι η γραμμική, η τύπου διαύλου, δακτυλίου, αστέρα, η κατανεμημένη, η πλήρως κατανεμημένη και η τύπου δένδρου.</p>
```

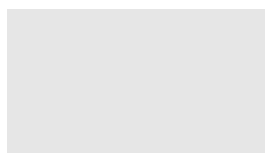
```
<span class="subtitle">Κατηγορίες Τοπολογιών</span></content>
```

Μία επιπλέον λειτουργία που θέλουμε να προσθέσουμε στις διδακτικές ενότητες είναι η επιλογή download του κειμένου που παρουσιάζεται. Προκειμένου να επιτύχουμε το σκοπό αυτό, δημιουργήσαμε pdf αρχεία, κάθε ένα από τα οποία περιέχει τη διδακτική ύλη κάθε μίας από τις διδακτικές ενότητες. Στη συνέχεια, σε κάθε ένα xml αρχείο προσθέσαμε την ακόλουθη εντολή:

```
<a href='arχειο.pdf' target='_blank'>Κατεβάστε το κείμενο σε pdf</a>
```

Στην ουσία δημιουργούμε ένα link, κάτω από τον τίτλο Κατεβάστε το κείμενο σε pdf το οποίο στοχεύει στο pdf αρχείο που έχουμε δημιουργήσει. Όταν ο χρήστης κάνει κλικ πάνω σε αυτό το link, τότε μπορεί να αποθηκεύσει στον υπολογιστή του το pdf αρχείο.

Μετά τη δημιουργία των xml αρχείων, επόμενο βήμα είναι να δημιουργήσουμε το φόντο στο οποίο θα εμφανίζονται, Εικόνα 6.90, καθώς και η ανάπτυξη του απαραίτητου κώδικα.



Εικόνα 6.90

Προκειμένου να φορτώσουμε στη σκηνή το αντίστοιχο xml αρχείο, αρχικά μετατρέπουμε το φόντο σε movie clip, και προσθέτουμε στο αντίστοιχο keyframe τον ακόλουθο κώδικα:

```
didaktikiEnotita1.load_xml("prox_page_1.xml");
```

Όπου `didaktikiEnotita(number)` είναι το αντίστοιχο movie clip (φόντο) στο οποίο θα εμφανιστεί το κείμενο, `load_xml()` είναι build in συνάρτηση του Actions Script για φόρτωση xml αρχείων, και μέσα στην παρένθεση γίνεται η δήλωση των αρχείων.

Για κάθε διδακτική ενότητα, βασισμένοι σε έρευνες για μαθησιακές δυσκολίες, αποφασίσαμε να δώσουμε στο χρήστη την επιλογή ακρόασης του κειμένου. Για αυτή την λειτουργία δημιουργήσαμε ένα κουμπί, Εικόνα 6.91, το οποίο επιλέγει ο χρήστης εάν θελήσει να ακούσει τις ηχογραφημένες παρουσιάσεις της διδακτικής ύλης (για κάθε μία από τις ενότητες έχουμε ηχογραφήσει ένα ξεχωριστό αρχείο).



Εικόνα 6.91

Εάν ο χρήστης κάνει κλικ στο κουμπί αυτό, τότε αρχίζει η αναπαραγωγή του αντίστοιχου μουσικού αρχείου. Για να σταματήσει την αναπαραγωγή, ο χρήστης κάνει δεύτερο κλικ στο κουμπί:

```
voiceButton.addEventListener(MouseEvent.CLICK, fl_ClickToPlayStopSound);
```

```
var fl_SC:SoundChannel;
```

```
var fl_ToPlay:Boolean = true;
```

```
function fl_ClickToPlayStopSound(evt:MouseEvent):void
```

```
{
```

```
if(fl_ToPlay)
```

```
{
```

```
var s:Sound = new Sound(new URLRequest("proxorimenosSections/proxorimenos/eisagogi.mp3"));
```

```
fl_SC = s.play();
```

```
}
```

```
else
```

```
{
```

```
fl_SC.stop();
```

```
}
```

```
fl_ToPlay = !fl_ToPlay;
```

```
}
```

Η προγραμματιστική τακτική που ακολουθήσαμε είναι αντίστοιχη με αυτήν της αναπαραγωγή της μουσικής που παρουσιάσαμε σε προηγούμενη ενότητα.

Υποεφαρμογές (παιχνίδι, τεστ quiz)

Κάθε ένα από τα frames που έχουμε καθορίσει, αντιστοιχεί στην αναπαραγωγή μίας συγκεκριμένης υποεφαρμογής. Για τον λόγο αυτόν, στο αντίστοιχο frame (με βάση των κουμπιών του μενού), φορτώνουμε στη σκηνή την αντίστοιχη υποεφαρμογή:

```
fl2_Loader.load(new URLRequest("efarmogi.swf"));
```

Η διαδικασία φόρτωσης αποτελείται από την εκτέλεση κλασικών εντολών για φόρτωση και προβολή flash εφαρμογών. Παρόλα αυτά, το κρίσιμο σημείο είναι η αφαίρεση των εφαρμογών από τον loader. Εάν δεν αφαιρέσουμε τις εφαρμογές από τον loader, τότε εάν ο χρήστης επιλέξει κάποια άλλη ενότητα από το μενού, η υποεφαρμογή συνεχίζει να τρέχει και να προβάλλεται στην οθόνη. Μία κλασική τακτική για να επιτύχουμε την αφαίρεση της υποεφαρμογής θα ήταν να προσθέσουμε μία εντολή `removeChild(loader)` για την αντίστοιχη υποεφαρμογή. Όμως αυτή η τακτική πρέπει να εφαρμοστεί όταν: 1) Ο loader δεν είναι κενός, 2) Η εφαρμογή πρέπει να αφαιρεθεί μόνο κατόπιν εντολής του χρήστη, δηλαδή αν αλλάξει ενότητα 3) Η εφαρμογή έχει προστεθεί στον loader. Εάν η εντολή `removeChild()` κληθεί σε οποιαδήποτε άλλη περίπτωση, τότε είτε θα καταλήξουμε στο να μην αναπαραχθεί ποτέ, είτε να προκαλέσουμε `compile errors`. Για να αποφύγουμε αυτά τα σφάλματα, πρέπει: 1) Να ελέγξουμε αν υπάρχει ο loader (αλλιώς θα προσπαθήσουμε να διαβάσουμε ένα `undefined object` το οποίο θα οδηγήσει σε `errors`), 2) Να ελέγξουμε ότι ο loader έχει κάποιο child (δεν είναι κενός), 3) Να ελέγξουμε πιο είναι το child που βρίσκεται στον loader, και 4) Να αφαιρέσουμε το συγκεκριμένο child. Βασισμένοι σε αυτά τα βήματα, προσθέσαμε το ακόλουθο κομμάτι κώδικα σε κάθε ένα από τα `keyframes`:

```
if(fl2_Loader) {  
  if(fl2_Loader.parent) {  
    fl2_Loader.unload();  
    removeChild(fl2_Loader);  
  }  
  else { trace("nothing");}  
}  
  
if(fl_Loader){  
  if(fl_Loader.parent){
```

```
fl_Loader.unload();  
removeChild(fl_Loader);  
}  
else { trace("nothing ");}  
}  
if(fll_Loader){  
if(fll_Loader.parent){  
fll_Loader.unload();  
removeChild(fll_Loader);  
}  
else { trace("nothing");}  
}
```

Στην ουσία αυτό που κάνουμε είναι να ελέγξουμε για την ύπαρξη του loader και εάν υπάρχει, στη συνέχεια να ελέγξουμε για το αντίστοιχο child. Εάν ο loader δεν υπάρχει, τότε δεν κάνουμε τίποτα. Κάνουμε έλεγχο για όλες τις υποεφαρμογές, κατά πόσο κάποια από αυτές αποτελεί child του loader. Αφαιρείται μόνο η εφαρμογή που έχει φορτωθεί, ενώ στις άλλες περιπτώσεις δεν εκτελούμε καμία ενέργεια.

Βίντεο

Για την αναπαραγωγή του βίντεο, στο αντίστοιχο keyframe, δημιουργούμε έναν media player, Εικόνα 6.92. Ο player αυτός παρέχεται από τη βιβλιοθήκη του Adobe Flash CS5, μαζί με built in συναρτήσεις.



Εικόνα 6.92

Στη συνέχεια, δίπλα από τον player δημιουργούμε ένα classic text στο οποίο θα προβάλλουμε την περιγραφή του βίντεο, Εικόνα 6.93.



Εικόνα 6.93

Για την αναπαραγωγή του βίντεο και την προβολή του κειμένου αναπτύξαμε τον ακόλουθο κώδικα:

```
videoPlayer.play();
```

Επικοινωνία

Για την επικοινωνία, αρχικά σχεδιάσαμε την φόρμα επικοινωνίας, Εικόνα 6.94

The image shows a contact form with the following fields and buttons:

- Name:** A text input field with a placeholder "Your Name".
- Email:** A text input field with a placeholder "Your Email".
- Subject:** A text input field with a placeholder "Your Subject".
- Message:** A large text area with a placeholder "Your Message".
- Buttons:** "Send" and "Reset" buttons.

Εικόνα 6.94

Κάθε μία από τις ετικέτες εισαγωγής, αποτελεί ένα object το οποίο θα χρησιμοποιήσουμε στο php script προκειμένου πατώντας το κουμπί «Send» ο χρήστης, να αποστέλλεται email με το περιεχόμενο του «Message»:

```
$contact_name = $_POST['name'];
```

```
$contact_email = $_POST['email'];
```

```
$contact_subject = $_POST['subject'];
```

```
$contact_message = $_POST['message'];
```

```
if( $contact_name == true )
```

```
{
```

```
    $sender = $contact_email;
```

```
    $receiver = "jasonmanouras@yahoo.com";
```

```
    $client_ip = $_SERVER['REMOTE_ADDR'];
```

```
    $email_body = "Name: $contact_name \nEmail: $sender \n\nSubject: $contact_subject \n\nMessage:
```

```
\n\n$contact_message \n\nIP: $client_ip \n\nFlash Contact Form From Educational Software Networks";
```

```
    $extra = "From: $sender\r\n" . "Reply-To: $sender \r\n" . "X-Mailer: PHP/" . phpversion();
```

```
    if( mail( $receiver, "Flash Contact Form - $contact_subject", $email_body, $extra ) )
```

```
{
```

```
        echo "success=yes";
    }
    else
    {
        echo "success=no";
    }
}
```

Κεφάλαιο 7. Συμπεράσματα

Στη πτυχιακή αυτή εργασία παρουσιάσαμε το εκπαιδευτικό λογισμικό το οποίο σχεδιάσαμε και αναπτύξαμε για το μάθημα των Δικτύων. Όπως παρουσιάσαμε, η διαδικασία ανάπτυξης ενός εκπαιδευτικού λογισμικού δεν αποτελεί μια απλή υπόθεση. Προκειμένου το εκπαιδευτικό λογισμικό που θα αναπτυχθεί να είναι αποδοτικό και να αποτελεί μία καλή και σωστή λύση εκπαίδευσης, πρέπει να ληφθούν υπόψιν τόσο θεωρίες για τη σωστή εκπαίδευση, τα άτομα στα οποία απευθύνεται το λογισμικό αυτό, καθώς και λύσεις που προσφέρονται προκειμένου το λογισμικό να είναι ελκυστικό στον εκπαιδευόμενο.

Το λογισμικό που αναπτύξαμε αποτελεί μία εκπαιδευτική λύση για μικτά περιβάλλοντα, υπό την έννοια ότι μπορεί να χρησιμοποιηθεί εξ'αποστάσεως (ο εκπαιδευόμενος και ο εκπαιδευτή να μην βρίσκονται στον ίδιο φυσικό χώρο), αλλά και μη. Επιπρόσθετα, το λογισμικό το οποίο παρουσιάσαμε αποτελείται από ένα διαδραστικό περιβάλλον, στο οποίο ο χρήστης έχει ελευθερία να επιλέξει τη διδακτική ύλη, καθώς και την πορεία κατά την πλοήγηση στο λογισμικό.

Η διδακτική διαδικασία που παρουσιάζουμε, δε βασίζεται μόνο πάνω σε κλασικές μεθόδους εκπαίδευσης, όπως είναι το κείμενο, αλλά σε μία πλήρως πολυμεσική (κείμενο, εικόνες, ήχοι, βίντεο) διδασκαλία. Ο εκπαιδευόμενος έχει τη δυνατότητα να διαβάσει τη διδακτική ύλη, καθώς και να αποθηκεύσει τα αντίστοιχα κείμενα στον υπολογιστή του. Επίσης λαμβάνουμε υπόψιν μας το σύνολο των εκπαιδευόμενων για τους οποίους η μέθοδος αυτή δεν αποτελεί τη βέλτιστη επιλογή, προσφέροντάς τους την επιλογή να ακούσουν αντί να διαβάσουν τη διδακτική ύλη. Επιπρόσθετα, στηριγμένοι σε μελέτες οι οποίες παρατηρούν ότι υπάρχει ομάδα ανθρώπων οι οποίοι απορροφούν καλύτερα γνώσεις εάν κατά τη διάρκεια του διαβάσματος αναπαράγεται μουσική, δίνουμε τη δυνατότητα στο χρήστη της επιλογής αναπαραγωγής μουσικής κατά τη διάρκεια της πλοήγησής του στο λογισμικό. Πέραν του κειμένου, των ηχογραφημένων ενοτήτων και της μουσικής συμπεριλαμβάνουμε εικόνες που απεικονίζουν έννοιες του κειμένου. Τέλος, δίνουμε τη δυνατότητα στο χρήστη να παρακολουθήσει βίντεο τα οποία προβάλλουν λειτουργίες που περιγράφονται στη διδαχθείσα ύλη.

Ένα άλλο ιδιαίτερο χαρακτηριστικό, του λογισμικού που παρουσιάζουμε, είναι η δυνατότητα που προσφέρουμε στο χρήστη να επιλέξει το γνωστικό του επίπεδο. Με αυτόν τον τρόπο, παρουσιάζουμε στον εκπαιδευόμενο διδακτική ύλη η οποία είναι πιο κοντά στις υπάρχουσες γνώσεις του, σε αντίθεση με το να τον βομβαρδίσουμε με εντελώς άγνωστες έννοιες --ή με έννοιες τις οποίες γνωρίζει. Με αυτόν τον τρόπο ομαλοποιούμε τη διαδικασία της εκπαίδευσης, φέρνοντάς την στα μέτρα του εκπαιδευόμενου.

Επιπρόσθετα, για κάθε διδακτική ενότητα, δίνουμε τη δυνατότητα στο χρήστη να λάβει μέρος σε τεστ τα οποία έχουμε αναπτύξει. Σε κάθε τεστ ο χρήστης καλείται να απαντήσει σε ερωτήσεις βασισμένες στις διδακτικές ενότητες. Στο τέλος του τεστ παρέχεται βαθμολογία

με το σκορ των σωστών και των λάθος απαντήσεων. Επιπλέον, βασισμένοι σε ερευνητικά αποτελέσματα τα οποία αποδεικνύουν τις ευεργετικές ιδιότητες των παιχνιδιών στην εκπαίδευση, αναπτύξαμε παιχνίδια και quiz για το εκπαιδευτικό μας λογισμικό. Στόχος των εφαρμογών αυτών είναι να συνδυάσουν τη διδακτική ύλη που παρουσιάζουμε, με πιο διασκεδαστικές μεθόδους.

Τέλος, λαμβάνουμε υπόψιν μας τη σημαντικότητα της επικοινωνίας του εκπαιδευόμενου με τον εκπαιδευτή, και παρέχουμε αυτή τη δυνατότητα στο χρήστη μέσω της φόρμας επικοινωνίας που αναπτύξαμε.

7.1 Μελλοντική Δουλειά

Καθώς η τεχνολογία αλλάζει, αλλάζουν και οι ανάγκες των λογισμικών. Για το εκπαιδευτικό λογισμικό που παρουσιάσαμε, θεωρούμε ότι μία πιθανή βελτίωση θα μπορούσε να είναι η ικανότητα live επικοινωνίας του εκπαιδευόμενου με τον εκπαιδευτή. Η επικοινωνία αυτή θα μπορούσε να έχει τόσο μορφή διδασκαλίας, όσο και μιας απλής κοινωνικής επαφής. Με τον τρόπο αυτό, το λογισμικό θα μπορούσε να πάρει μία πιο προσωποποιημένη μορφή.

Το εκπαιδευτικό λογισμικό το οποίο αναπτύξαμε, είναι εξολοκλήρου βασισμένο στη τεχνολογία Flash, η οποία παρέχει τη δυνατότητα αναπαραγωγής του τόσο σε online όσο και σε offline περιβάλλοντα. Η τεχνολογία αυτή είναι ευρέως διαδεδομένη, και εύκολα προσβάσιμη. Παρόλα αυτά, τελευταία έχει παρατηρηθεί μία στροφή στην HTML5, η οποία μπορεί να παρέχει τα ίδια χαρακτηριστικά με τη Flash. Μεγάλες εταιρίες όπως η Apple, αλλά ανοιχτές ομάδες ανάπτυξης λογισμικού, όπως η Linux, πρόσφατα ανακοίνωσαν την αντικατάσταση της Flash από τη HTML5. Για το λόγο αυτόν στηρίζουμε την άποψη ότι μια

άλλη μελλοντική βελτίωση για το λογισμικό μας είναι η ενσωμάτωσή του στην τάση που ορίζει την αντικατάσταση της Flash απο τη HTML5.

Κώδικας

Τεστ

```
var qno=0;var rnd1; var rnd2;

tick.visible=false;cross.visible=false;

var right_answers=0;var wrong_answers=0;

function change_question(){

    if(tick.visible){right_answers++;}

    if(cross.visible){wrong_answers++;}

    if(qno==questions.length){gotoAndPlay(2);}else{

        tick.visible=false;cross.visible=false;

        rnd1=Math.ceil(Math.random()*3);

        rnd2=Math.ceil(Math.random()*questions.length)-1;

        q.text=questions[rnd2];

        if(questions[rnd2]=="x"){change_question();}

        questions[rnd2]="x";

        enable_disable(1);

        if(rnd1==1){opt1.text=answers[rnd2][0];opt2.text=answers[rnd2][1];opt3.text=answers[rnd2][2];}

        if(rnd1==2){opt1.text=answers[rnd2][2];opt2.text=answers[rnd2][0];opt3.text=answers[rnd2][1];}

        if(rnd1==3){opt1.text=answers[rnd2][1];opt2.text=answers[rnd2][2];opt3.text=answers[rnd2][0];}

    }}

function enable_disable(a){

    if(a==0){shade1.mouseEnabled=false;shade2.mouseEnabled=false;shade3.mouseEnabled=false;}

    if(a==1){shade1.mouseEnabled=true;shade2.mouseEnabled=true;shade3.mouseEnabled=true;}}

change_question();
```

```

next_b.addEventListener(MouseEvent.CLICK, ButtonAction1);

function ButtonAction1(eventObject:MouseEvent) {qno++;change_question();}

shade1.addEventListener(MouseEvent.CLICK, ButtonAction2);

shade2.addEventListener(MouseEvent.CLICK, ButtonAction3);

shade3.addEventListener(MouseEvent.CLICK, ButtonAction4);

function ButtonAction2(eventObject:MouseEvent)

{enable_disable(0);if(rnd1==1){tick.visible=true;tick.y=shade1.y}else{cross.visible=true;cross.y=shade1.y}}

function ButtonAction3(eventObject:MouseEvent)

{enable_disable(0);if(rnd1==2){tick.visible=true;tick.y=shade2.y}else{cross.visible=true;cross.y=shade2.y}}

function ButtonAction4(eventObject:MouseEvent)

{enable_disable(0);if(rnd1==3){tick.visible=true;tick.y=shade3.y}else{cross.visible=true;cross.y=shade3.y}}

stop();

```

Κρεμάλα

```

package {

import flash.display.MovieClip;

import flash.events.MouseEvent;

import flash.events.KeyboardEvent;

public class Main extends MovieClip {

public var wordlist:WordList;

public var CurrentWord:String;

public var CurrentClue:String;

public var rnd:int;

public var Letters:Array=new Array;

public var i:int;

public var tile:Tile;

```

```

public var TileArray:Array=new Array;

public var UserKeyPress:String;

public var Alphabet:Array=new Array;

public var Found:Boolean;

public var Win:Boolean;

public var Lose:Boolean;

public var DefeatCounter:int;

public var UsedCharacters:Array=new Array;

public var FirstPlayThrough:Boolean= true;

public function Main() {

stop();

wordlist=new WordList;

PlayButton.addEventListener(MouseEvent.CLICK, ButtonAction1);

function ButtonAction1(eventObject:MouseEvent) {gotoAndStop(2); SetUp();}

}

public function SetUp(){

DefeatCounter=0;

Man.gotoAndStop(1);

trace(wordlist.Words);trace(wordlist.Clues);

ChooseAWord();

SplitIntoLetters();

Display();

AddControls();

}

public function ChooseAWord(){

rnd=Math.random()*wordlist.Words.length;

CurrentWord=wordlist.Words[rnd];

CurrentClue=wordlist.Clues[rnd];

```

```

trace(CurrentWord);trace(CurrentClue);

}

public function SplitIntoLetters(){
Letters=CurrentWord.split("");
trace(Letters);
}

public function Display(){
MessageBoard.visible=false;
DisplayClue.text=CurrentClue;//embed chars
for(i=0;i<Letters.length;i++){
tile=new Tile;addChild(tile);
tile.x=200+(35*i);tile.y=250;
TileArray[i]=tile;
}
}

public function AddControls(){
if(FirstPlayThrough){
FirstPlayThrough=false;
stage.focus=stage;
stage.addEventListener(KeyboardEvent.KEY_UP,keyUpListener);
function keyUpListener(e:KeyboardEvent):void {trace(e.keyCode);
UserKeyPress=ConvertToLetter(e.keyCode);
ProcessAnswer(UserKeyPress);
}}}

public function ConvertToLetter(a){
Alphabet=["α","β","γ","δ","ε","ζ","η","θ","ι","κ","λ","μ","ν","ξ","ο","π","ρ","σ","τ","υ","φ","χ","ψ","ς","ω"];
if(a>64 && a<89){
return(Alphabet[a-65]);
}
}
}

```



```

else{return("");}

} //convert

public function ProcessAnswer(a){

if(a!="" && !Lose && !Win){

Found=false;

for (i=0;i<Letters.length;i++){

if(a==Letters[i]){TileArray[i].Blank.text=a;Found=true;}

} // for

if(Found){CheckForWin();}

if(!Found && Unused(a)){AddBodyPart();

DisplayWrongLetter(a);

CheckForDefeat();

} //if !found

} //if a!=""...

else{

if(Win || Lose){RestartGame();}

} //else

} //processanswer

public function CheckForWin(){

Win=true;

for (i=0;i<Letters.length;i++){

if(TileArray[i].Blank.text!=Letters[i]){Win=false;}

} //for i

if(Win){MessageBoard.visible=true;

MessageBoard.gotoAndStop(1);

} //if win

} //Check for win

```

```

public function Unused(a){
for (i=0;i<UsedCharacters.length;i++){
if(UsedCharacters[i]==a){return false;}
};//for i
return true;
};//unused

public function AddBodyPart(){
Man.nextFrame();
};//addbodypart

public function CheckForDefeat(){
DefeatCounter++;
if(DefeatCounter==6){Lose=true;
MessageBoard.visible=true;
MessageBoard.gotoAndStop(2);}
};//check for defeat

public function DisplayWrongLetter(a){
UsedCharacters[UsedCharacters.length]=a;
BadLetters.text="";
for(i=0;i<UsedCharacters.length;i++){
BadLetters.appendText(UsedCharacters[i]+" ");
};//fori
};//displaywrongletter

public function RestartGame(){
Win=false;Lose=false;BadLetters.text="";
for(i=0;i<Letters.length;i++){

```

```

removeChild(TileArray[i]);TileArray[i]=null;

} //for i

Letters=[];UsedCharacters=[];TileArray=[];

SetUp();

} //restartgame

}}

```

```

package {

import flash.display.MovieClip;

public class WordList extends MovieClip {

public var Words:Array = new Array;

public var Clues:Array = new Array;

public function WordList() {

Words=["αστερας","χερτζ","ναυπακτος","ραμ","συχνοτητα","σημα"];

Clues=["είδος τοπολογίας","μονάδα συχνότητας","τόπος","μέρος υπολογιστή","κύμα","θόρυβος"]

} //wl

}}

```

Quiz

```

package {

import flash.display.MovieClip;

import flash.events.MouseEvent;

public class Game extends MovieClip {

public var topology_list:Array;

public var display_list:Array;

public var topology_objects:Array;

```

```

        public var buttons:Array;

        public var num_items:int;

        public var i:int;var ii:int;

        public var temp1:String;var temp2:int;var rnd1:int;var rnd2:int;

        public var namebox:Namebox;

        public var answer1:String;

        public var answer2:String;

        public var num_correct:int;

        public var openpix:Openpix;

        public var frame:Frame;

        public var sound1:Sound1= new Sound1;

public var sound2:Sound2= new Sound2;

public function Game() {

trace("This quiz is property of Jason Manouras");

startscreen();

public function startscreen(){

gotoAndStop(1);

bplay.addEventListener(MouseEvent.CLICK,Nav1);

function Nav1(eventObject:MouseEvent){gotoAndStop(2); populate();}

    }

public function populate(){

num_correct=0;

answer1="";answer2="";

topology_list=["star","bus","linear","ring","tree"];

display_list=new Array;buttons=new Array;

for(i=0;i<topology_list.length;i++){display_list[i]=topology_list[i];}

scramble(display_list);trace(topology_list);trace(display_list);

for(i=0;i<topology_list.length;i++){

```

```

namebox=new Namebox;addChild(namebox);

namebox.topologyname.text=display_list[i];

namebox.mouseChildren = false;

namebox.x=50+(i*120);namebox.y=350;

buttons[i]=namebox;

frame=new Frame;addChild(frame);frame.visible=false;

getimages();

public function scramble(a){
for(i=0;i<30;i++){
rnd1=Math.random()*a.length;rnd2=Math.random()*a.length;
temp1=a[rnd1];a[rnd1]=a[rnd2];a[rnd2]=temp1;

public function getimages(){
topology_objects=new Array;
for(i=0;i<topology_list.length;i++){
openpix=new Openpix;addChild(openpix);openpix.getpix(topology_list[i],70+(i*110),120);
topology_objects[i]=openpix;

addListeners();

public function addListeners(){
for(i=0;i<topology_objects.length;i++){
topology_objects[i].mouseChildren = false;
topology_objects[i].addEventListener(MouseEvent.CLICK, pic_click);

for(i=0;i<display_list.length;i++){
buttons[i].addEventListener(MouseEvent.CLICK, but_click);

```

```

public function pic_click(event:MouseEvent){

frame.visible=true; frame.x=event.target.x;frame.y=event.target.y;

trace(event.target.pic_label);

answer1=event.target.pic_label;

    if(answer1==answer2){correct();}else{notcorrect();}

    }

    public function but_click(event:MouseEvent){

        for(i=0;i<buttons.length;i++){buttons[i].gotoAndStop(1);}

        trace(event.target.topologyname.text);

        event.target.gotoAndStop(2);

        answer2=event.target.topologyname.text;

        if(answer1==answer2){correct();}else{notcorrect();}

    }//but_click

    public function correct(){num_correct++;sound1.play();

        for(i=0;i<buttons.length;i++){if(buttons[i].topologyname.text==answer2){break;}}

        for(ii=0;ii<topology_list.length;ii++){if(topology_list[ii]==answer2){break;}}

            buttons[i].x=topology_objects[ii].x-5;

            buttons[i].y=topology_objects[ii].y+110;

            buttons[i].gotoAndStop(1);

            buttons[i].removeEventListener(MouseEvent.CLICK, but_click);

            topology_objects[ii].removeEventListener(MouseEvent.CLICK, pic_click);

            if(num_correct==topology_objects.length){congrats.text="Συγχαρητήρια!";}

            answer1="";answer2="";

    }//correct

    public function notcorrect(){

        if(answer1!="" && answer2!=""){

            sound2.play();

```

```

        answer1="";answer2="";

        frame.visible=false;

        for(i=0;i<display_list.length;i++){buttons[i].gotoAndStop(1);}//fori
package {

    import flash.display.MovieClip;

    import flash.net.URLLoader;

    import flash.net.URLRequest;

    import flash.display.Loader;

    public class Openpix extends MovieClip{

        var imageLoader:Loader;

        var pic_label:String;

        public function Openpix() {

            // constructor code

        }

        public function getpix(a,mcx,mcy){

            pic_label=a;

            var imageLoader:Loader = new Loader();

            var image:URLRequest = new URLRequest(a+".jpg");

            imageLoader.load(image);

            addChild (imageLoader);

            this.x=mcx;

            this.y=mcy;

            //imageLoader.x = mcx;

            //imageLoader.y = mcy;

        }//openpix

    }

```

Προβολή Βοήθειας

```
function fl2_ClickToPosition(event:MouseEvent):void

{if (fl2_ToLoad){

fl2_TF = new TextField();

fl2_TF.autoSize = TextFieldAutoSize.LEFT;

fl2_TF.background = true;

fl2_TF.backgroundColor= 0x000000;

fl2_TF.border = true;

fl2_TF.wordWrap = true;

// fl2_TF.fontStyle = BOLD;

fl2_TF.textColor = 0xFF0000;

fl2_TF.width = 650;

fl2_TF.height = 140;

fl2_TF.x = 30;

fl2_TF.y = 450;

fl2_TF.text = fl2_TextToDisplay;

addChild(fl2_TF);

fl2_ToLoad = false;

}

else

{

fl2_ToLoad = true

removeChild(fl2_TF);

}}
```


Αναπαραγωγή μουσικής

```
var fl_SC:SoundChannel;

var fl_ToPlay:Boolean = true;

function fl_ClickToPlayStopSound(evt:MouseEvent):void
{
    if(fl_ToPlay)
    {
        var s:Sound = new Sound(new URLRequest("music/mousiki.mp3"));
        fl_SC = s.play();
    }
    else
    {
        fl_SC.stop();
    }
    fl_ToPlay = !fl_ToPlay;
}
```

Κουμπί Πίσω

```
backButton.addEventListener(MouseEvent.CLICK, f3_ClickToLoadUnloadSWF);

var f3_ToLoad:Boolean = true;

var f3_Loader:Loader;

function f3_ClickToLoadUnloadSWF(event:MouseEvent):void
{
    if(f3_ToLoad)
    {
        SoundMixer.stopAll();

        f3_Loader = new Loader();

        f3_Loader.load(new URLRequest("intro.swf"));
    }
}
```

```

        f3_Loader.x=Xpos;

        f3_Loader.y=Ypos;

        addChild(f3_Loader);

    }

    else

    {

        f3_Loader.unload();

        removeChild(f3_Loader);

        f3_Loader = null;

    }

    // Toggle whether you want to load or unload the SWF

    f3_ToLoad = !f3_ToLoad;

}

```

Skip Intro Κουμπι

```

skipIntro.addEventListener(MouseEvent.CLICK, f);

```

```

function f(event:MouseEvent):void{

    SoundMixer.stopAll();

        fl_Loader.unload();

        removeChild(fl_Loader);

        fl_Loader.load(new URLRequest("choose.swf"));

        fl_Loader.x=Xpos;

        fl_Loader.y=Ypos;

        addChild(fl_Loader);

}

```

Φόρτωση Flash στην κυρίως σκηνή

```
function fl_ClickToLoadUnloadSWF(event:MouseEvent):void
{
    if(fl_ToLoad)
    {
        fl_Loader = new Loader();
        fl_Loader.load(new URLRequest("arxeio.swf"));
        fl_Loader.x=Χπος;
        fl_Loader.y=Υπος;
        addChild(fl_Loader);
    }
    else
    {
        fl_Loader.unload();
        removeChild(fl_Loader);
        fl_Loader = null;
    }
    fl_ToLoad = !fl_ToLoad;
}
```

Έλεγχος και Αφαίρεση Φορτωμένων Υποεφαρμογών

```
if(fl2_Loader) {
    if(fl2_Loader.parent) {
        fl2_Loader.unload();
        removeChild(fl2_Loader);}
    }
    else { trace("nothing");}
}
```

```
if(fl_Loader){
if(fl_Loader.parent){

fl_Loader.unload();
removeChild(fl_Loader);
}
else { trace("nothing ");}
}
if(fll_Loader){
if(fll_Loader.parent){

fll_Loader.unload();
removeChild(fll_Loader);
}
else { trace("nothing");}
}}
```

Επικοινωνία

```
$contact_name = $_POST['name'];
$contact_email = $_POST['email'];
$contact_subject = $_POST['subject'];
$contact_message = $_POST['message'];
```

```
if( $contact_name == true )
{
    $sender = $contact_email;
    $receiver = "jasonmanouras@yahoo.com";
    $client_ip = $_SERVER['REMOTE_ADDR'];
```

```
$email_body = "Name: $contact_name \nEmail: $sender \n\nSubject: $contact_subject \n\nMessage:  
\n\n$message \n\nIP: $client_ip \n\nFlash Contact Form From Educational Software Networks";
```

```
$extra = "From: $sender\r\n" . "Reply-To: $sender \r\n" . "X-Mailer: PHP/" . phpversion();
```

```
if( mail( $receiver, "Flash Contact Form - $contact_subject", $email_body, $extra ) )
```

```
{
```

```
    echo "success=yes";
```

```
}
```

```
else
```

```
{
```

```
    echo "success=no";
```

```
}
```

```
}
```

Βιβλιογραφία

Ελληνική

- ΕΑΙΤΥ (2007). Επιμορφωτικό υλικό για την εκπαίδευση των επιμορφωτών στα ΠΑΚΕ. Πάτρα
- ΙΤΥ (2002). Εκπαιδευτικό λογισμικό: Πρώτη γνωριμία με διαθέσιμο εκπαιδευτικό λογισμικό, στο πλαίσιο του έργου: Επιμόρφωση των εκπαιδευτικών στην αξιοποίηση των ΤΠΕ στην εκπαίδευση. Πάτρα
- Μικρόπουλος Α.(2000). Εκπαιδευτικό Λογισμικό. Θέματα Σχεδίασης και αξιολόγησης λογισμικού υπερμέσων. Εκδόσεις Κλειδάριθμος. Αθήνα.
- Παπαδόπουλος, Γ. (2001). Έλεγχος ποιότητας εκπαιδευτικού λογισμικού: Ο σχεδιασμός του Παιδαγωγικού ινστιτούτου. Εισήγηση στο 1ο Συνέδριο Σύρου Οι ΤΠΕ στην εκπαίδευση
- Παπαδόπουλος Γ. Χρ., Πολάτογλου Χ.(2002), Αρχές σχεδίασης εκπαιδευτικού λογισμικού και διερεύνηση ένταξής του στη διδακτική διδασκαλία. 3ο Πανελλήνιο Συνέδριο Διδακτική των Φυσικών Επιστημών και Εφαρμογή Νέων Τεχνολογιών στην Εκπαίδευση.
- ΠΙ (2007). Επιμορφωτικό υλικό Γενικού Μέρους του Προγράμματος Σπουδών για την εκπαίδευση των επιμορφωτών. Αθήνα
- Σολομωνίδου Χ., (2003), Σύγχρονη Εκπαιδευτική Τεχνολογία. Υπολογιστές και μάθηση στην κοινωνία της γνώσης. Εκδόσεις Κώδικας. Θεσσαλονίκη.
- Σταχτέας Χ.,(2002). Πληροφορική στην Εκπαίδευση. Οι υπολογιστές στο σχολείο του μέλλοντος. Εκδόσεις Τυπωθωτά Γιώργος Δαρδανός, Αθήνα.
- Ψυχάρης Σαράντος & Φραγκάκης Κυπριανός, (2006).«Η χρήση των Τεχνολογιών Πληροφορίας και Επικοινωνιών για την ανάπτυξη μεταγνωστικών δεξιοτήτων και η χρήση διαδικτυακού εργαλείου για την αξιολόγησή τους. Συνέδριο της ΕΤΠΕ (Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας και Επικοινωνιών στην Εκπαίδευση). Θεσσαλονίκη 5-8 Οκτωβρίου 2006. Σελ.523-530.

Αγγλική

- Conati, C., & Zhou, X. (2002). Modeling students' emotions from cognitive appraisal in educational games. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Proceedings of the Intelligent Tutoring Systems 2002*, Lecture Notes in Computer Science, 2363, Berlin Heidelberg: Springer, 944-954.
- Bates, T.(1995). Technology, Open Learning and Distance Education. New York: Routledge
- BECTA (2007). Enhancing Learning; virtual worlds, simulations and game based learning. Presentation .
- Garris, R., Ahlers, R. and Driskell, J. (2002). Games, motivation and learning: A research and practice model. *Simulation and Gaming*, 33(4). 441-467.

- Jayne M. Standley. (1996). A Meta-Analysis on the Effects of Music as Reinforcement for Education/Therapy Objectives. *Journal of Research in Music Education*. 44(2): 105-133.
- Johnson, S. (2005). Everything bad is good for you: How Today's Popular Culture Is Actually Making Us Smarter. London: Allen Lane.
- Kitao, K., Kitao, S. K. (1997) "Selecting and developing teaching/learning materials." *The Internet TESL Journal*, vol. 4, no. 4
- Lautillard D. (1993). Rethinking University Teaching: A framework for the effective use of educational technology. Routledge. London.
- L.C.Schaffer, M.J. Hannafin (1993). The effects of progressive interactivity on learning from interactive video. In *Educational Technology Research and Development* 34(2):89-96
- McFarlane, A., Sparrowhawk, A. and Heald Y.(2002). Report on the educational use of computer games. Teachers Evaluating Educational Multimedia report.
- Papert, S. (1993). *The Children's Machine: Rethinking School in the Age of the Computers*, New York: Basic Books
- Schlosser, C. A., and Anderson, M. L. *Distance Education: Review of the Literature*. Washington, DC: Association for Educational Communications and Technology, 1994. (ED 382 159)
- Sherry, L.(1966). "Issues in Distance Learning." *International Journal of Educational Telecommunications* 1, no.4 : 337-365 .
- Stubbs, M. *Language. (1976). Schools, and Classrooms*. London: Methuen.
- Timothy B. Lee. (2010). *The Death of Flash and the Quiet Triumph of Open Standards*.
- Vaughan-Nichols, S.J. (2010). Will HTML5 Standardize the web?. *IEEE Computer* 43(4): 13-15 (2010)
- Virvou, M., Katsionis, G., & Manos, K. (2005). Combining Software Games with Education: Evaluation of its Educational Effectiveness. *Educational Technology & Society*, 8 (2), 54-65.
- Virvou, M., & Alepis, E., (2005). Mobile educational features in authoring tools for personalised tutoring. In *Computers & Education*, Vol 44, No. 1, pp. 53-68, Elsevier Science.
- Virvou, M., Katsionis, G.: On the usability and likeability of virtual reality games for education: The case of VR-ENGAGE. *Computers and Education* 50(1): 154-178 (2008)
- Virvou M., Katsionis G., Manos K., 2005, Combining Software games with

educations: Evaluation of its educational effectiveness.

Weinstein, P. (1997). Education Goes the Distance. *Technology and Learning* 17, no. 8:24-25.

Woodhouse David, McDougall Anne (1986). *Computers: Promise and Challenge in Education*.