



Α.Τ.Ε.Ι ΚΡΗΤΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΠΟΛΥΜΕΣΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ:

Σχεδίαση 2D παιχνιδιού σε Adobe Flash

Σπουδαστής:

Μπροφίδης Νικόλαος

Εισηγητής:

Μαλάμος Αθανάσιος

Περιεχόμενα

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ	1
Πρόλογος	4
Κεφ. 1 - Η έννοια των πολυμέσων.....	5
1.1 Εισαγωγή	5
1.2 Ετυμολογία	5
1.3 Ορισμός	6
1.4 Χαρακτηριστικά	8
1.5 Αυτόνομα και δικτυωμένα πολυμέσα	8
1.6 Φιλοσοφία των Hypertext και Hypermedia	9
1.7 Ιστορική αναδρομή-Υπάρχοντα συστήματα	11
1.8 Βασικά δομικά στοιχεία	15
1.8.1 Κόμβοι.....	16
1.8.2 Σύνδεσμοι.....	16
1.8.3 Δομικά μοντέλα hypertext.....	17
1.8.4 Περιήγηση (navigation) και περιόδευση (browsing)	18
1.8.5 Browsers.....	20
1.9 Hypermedia : Μέλλον και προοπτικές	22
Κεφ. 2 - Το πρόγραμμα Macromedia Flash.....	24
2.1 Εισαγωγή	24
2.2 Απλές σημειώσεις για το Flash	24
Κίνηση και επίπεδα	25
Σύμβολα και στιγμιότυπα.....	27
Κουμπιά.....	29
2.3 Παράδειγμα	29
Κεφ. 3 – Παιχνίδια που υλοποιήθηκαν.....	31
3.1 Ανάλυση και επεξήγηση της εφαρμογής Avoid the cars	31
3.2 Ανάλυση και επεξήγηση των επιλογών της εφαρμογής Snakes	32
Παράρτημα Α - Παιχνίδι: Avoid the Cars	33
Επίπεδο «lv1» (Frames 1 – 4)	33

Επίπεδο «lv2» (Frames 5 – 9)	37
Επίπεδο «lv3» (Frames 10 – 14)	37
Επίπεδο «lv4» (Frames 15 – 19)	37
Επίπεδο «Win» (Frames 20 -25)	38
Επίπεδο «Game Over» (Frames 25 -31)	38
Υπολογισμός «Total Score»	39
Κίνηση Παιδιού	39
Παράρτημα Β - Παιχνίδι: Snake.....	40
Frames “1-2-3” Level 1	40
Frames “4-5-6” Level 2	45
Frames 7-8-9 “Level 3”	49
Frame 10 Game Over	52
Achievement (To score)	53

Πρόλογος

Σκοπός της παρούσας πτυχιακής εργασίας είναι η σχεδίαση, η ανάπτυξη και η υλοποίηση 2 πολυμεσικών εφαρμογών (2 δισδιάστατων παιχνιδιών) στο πρόγραμμα Adobe Flash της Macromedia.

Καταρχάς θα αναλύσουμε τα θέματα που αφορούν την έννοια του μέσου και τις πολυμεσικές εφαρμογές κάνοντας μια εισαγωγή σε αυτά και αναλύοντας τα σαν ορολογία αλλά και ετυμολογικά. Στη συνέχεια θα αναφερθούμε γενικά για το πρόγραμμα που χρησιμοποιήσαμε για την υλοποίηση των 2 παιχνιδιών.

Για την υλοποίηση αυτής των πολυμεσικών εφαρμογών εργασθήκαμε σε περιβάλλον Flash 8 της Macromedia και ο βασικός κώδικας της υλοποίησης που χρησιμοποιήσαμε είναι Action Script. Θα αναφέρουμε τον τρόπο υλοποίησης της παραπάνω εφαρμογών, τα βασικά σημεία λειτουργίας τους καθώς και τον τρόπο χρήσης τους.

Κεφ. 1 - Η έννοια των πολυμέσων

1.1 Εισαγωγή

Σε αυτή την ενότητα της πτυχιακής θα γίνει μια σύντομη εισαγωγή στον κόσμο των πολυμέσων, θα αναζητήσουμε την ετυμολογία τους. Στην συνέχεια θα δώσουμε έναν ορισμό τους και θα περιγράψουμε τα χαρακτηριστικά ενός συστήματος πολυμέσων.

Τα πολυμέσα είναι μία από τις πιο πολυσυζητημένες τεχνολογίες των αρχών της δεκαετίας του 90. Το ενδιαφέρον αυτό είναι απόλυτα δικαιολογημένο, αφού τα πολυμέσα αποτελούν το σημείο συνάντησης πέντε μεγάλων βιομηχανιών: της πληροφορικής, των τηλεπικοινωνιών, ηλεκτρονικών εκδόσεων, της βιομηχανίας audio και video καθώς και της βιομηχανίας της τηλεόρασης και του κινηματογράφου.

Μια ανάλογη αναστάτωση επέφερε και η εμφάνιση της επιστήμης των δικτύων υπολογιστών στη δεκαετία του 70, φέρνοντας πιο κοντά την πληροφορική με τις τηλεπικοινωνίες. Αυτή η προσέγγιση οδήγησε σε προϊόντα που στόχευαν κυρίως στην αγορά των επιχειρήσεων. Τα πολυμέσα έκαναν κάτι περισσότερο, διεύρυναν την αγορά των προϊόντων των παραπάνω βιομηχανιών που πλέον στοχεύουν και στους καταναλωτές.

Η πληθώρα και οι ποικιλία των νέων προϊόντων καθώς και η προσπάθεια εκμετάλλευσης του ενδιαφέροντος που επέδειξε το αγοραστικό κοινό για την τεχνολογία των πολυμέσων συνετέλεσαν στην σύγχυση που υπάρχει ακόμα και σήμερα όσον αφορά στο τι είναι και τι δεν είναι ένα σύστημα πολυμέσων.

Μια καλή αρχή για τον καθορισμό του όρου είναι η ανάλυση της ετυμολογίας του.

1.2 Ετυμολογία

Ο αγγλικός όρος, που εδώ έχει αποδοθεί ως πολυμέσα, είναι **multimedia**. Ο όρος αυτός αποτελείται από δύο μέρη: το πρόθεμα **multi** και τη ρίζα **media**.

Multi: προέρχεται από τη λατινική λέξη **multus** και σημαίνει "πολύαριθμος", "πολλαπλός".

Media: είναι ο πληθυντικός αριθμός της επίσης λατινικής λέξης **medium** που σημαίνει "μέσο", "κέντρο". Πιο πρόσφατα η λέξη **medium** άρχισε να χρησιμοποιείται και ως "ενδιάμεσος", "μεσολαβητής".

Κατά συνέπεια ο ορισμός που προκύπτει είναι:

Multimedia σημαίνει "πολλαπλοί μεσολαβητές" ή "πολλαπλά μέσα" και χρησιμοποιείται είτε ως ουσιαστικό είτε ως επίθετο.

1.3 Ορισμός

Η πρώτη προσέγγιση του ορισμού δεν μας λέει και πολλά πράγματα. Μπορούμε όμως να τον βελτιώσουμε αναλογιζόμενοι τον τρόπο χρήσης των όρων *multi* και *media*. Ο αγγλικός όρος *media* χρησιμοποιείται σε πολλούς οικονομικούς, τεχνικούς και επιστημονικούς τομείς με διαφορετικές σημασίες. Το κοινό σημείο αυτών των χρήσεων είναι ότι σχετίζονται πάντοτε με κάποιο είδος χειρισμού πληροφορίας:

- Αποθήκευση και επεξεργασία στην πληροφορική
- Παραγωγή στον χώρο των εκδόσεων
- Διανομή στον χώρο των μαζικών μέσων επικοινωνίας
- Μετάδοση στις τηλεπικοινωνίες
- Αντίληψη κατά την αλληλεπίδραση του ανθρώπου με το περιβάλλον του.

Κατά συνέπεια μπορούμε να βελτιώσουμε τον ορισμό ως εξής:

Πολυμέσα στον χώρο της τεχνολογίας πληροφορίας (**information field**) σημαίνει “πολλαπλοί μεσολαβητές” μεταξύ της πηγής και του παραλήπτη της πληροφορίας ή “πολλαπλά μέσα” μέσω των οποίων η πληροφορία αποθηκεύεται, μεταδίδεται, παρουσιάζεται ή γίνεται αντιληπτή..

Σύμφωνα με αυτόν τον ορισμό, ένα σύστημα που συνδυάζει, για παράδειγμα, τον έλεγχο βιντεοκασέτας και οπτικών μέσων αποθήκευσης μπορεί να χαρακτηριστεί ως σύστημα πολυμέσων. Επίσης συστήματα πολυμέσων θα είναι η εφημερίδα, που συνδυάζει κείμενο και εικόνα, και η τηλεόραση, που συνδυάζει ήχο και κινούμενη εικόνα. Εδώ δεν αναφερόμαστε σε τόσο ευρύ φάσμα συστημάτων. Περιοριζόμαστε σε αυτά στα οποία η πληροφορία είναι ψηφιακή (ή ψηφιοποιημένα - digitized) και ελέγχεται από υπολογιστή. Ενδιαφερόμαστε δηλαδή για ψηφιακά πολυμέσα τα οποία και ορίζουμε ως εξής:

“Ψηφιακά πολυμέσα είναι ο τομέας που ασχολείται με την ελεγχόμενη από υπολογιστή ολοκλήρωση κειμένου, γραφικών, ακίνητης και κινούμενης εικόνας, animation, ήχου, και οποιουδήποτε άλλου μέσου ψηφιακής αναπαράστασης, αποθήκευσης, μετάδοσης και επεξεργασίας της πληροφορίας”

Επειδή στη συνέχεια θα ασχοληθούμε μόνο με τα ψηφιακά πολυμέσα, θα χρησιμοποιούμε τον όρο πολυμέσα εννοώντας τα ψηφιακά πολυμέσα. Επίσης ως μέσο θα εννοούμε τους τύπους πληροφορίας που αναφέρει ο παραπάνω ορισμός.

Διαβάζοντας αυτόν τον ορισμό, δημιουργείται το ερώτημα: ποιους και πόσους τύπους πληροφορίας πρέπει να συνδυάζει ένα σύστημα, για να μπορεί δίκαια να χαρακτηρίζεται ως σύστημα πολυμέσων; Όπως είναι φανερό, η απάντηση σε αυτό το ερώτημα δεν μπορεί να είναι αυστηρή, γιατί δεν έχει γίνει κάποια συμφωνία πάνω στον ορισμό των πολυμέσων. Όμως στην πράξη έχουν δημιουργηθεί κάποιοι *de facto* κανόνες που

καθορίζουν τι πρέπει να περιλαμβάνει ένα σύστημα πολυμέσων ανάλογα με το είδος της εφαρμογής. Σαν κατευθυντήρια γραμμή μπορούμε να δώσουμε τον παρακάτω ορισμό:

Στην πράξη, ο όρος πολυμέσα υπονοεί την ολοκλήρωση ενός τουλάχιστον “διακριτού” τύπου πληροφορίας και ενός “συνεχούς”.

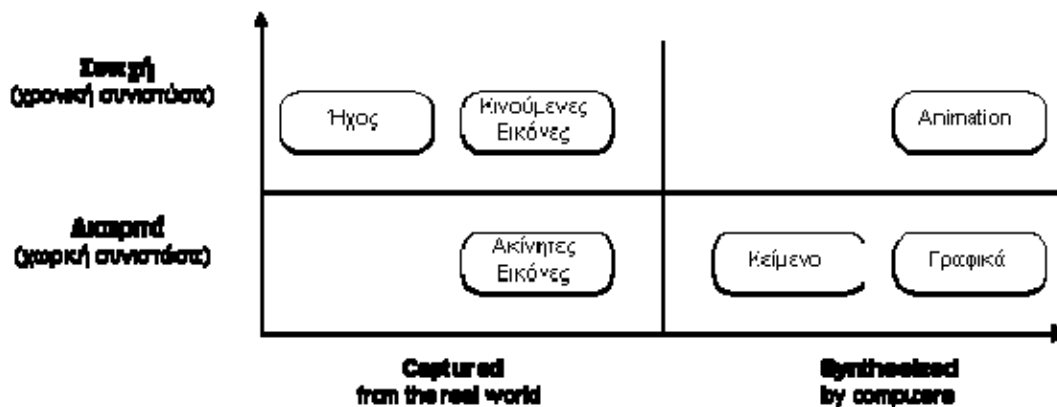
Στον παραπάνω ορισμό έχει γίνει διαχωρισμός των τύπων πληροφορίας σε διακριτούς και συνεχείς. Ένας άλλος διαχωρισμός είναι σε **captured** και **synthesized** μέσα. Ας δούμε τι σημαίνουν αυτοί οι διαχωρισμοί (Σχήμα 1):

· *Captured versus synthesized media*

Αυτός ο διαχωρισμός αναφέρεται στον τρόπο μεταφοράς της πληροφορίας στη μορφή που υπαγορεύει ο κάθε τύπος. Αν η πληροφορία συλλαμβάνεται απευθείας από τον πραγματικό κόσμο μιλάμε για captured media ενώ αν δημιουργείται από τον άνθρωπο μέσω κάποιων εργαλείων έχουμε τα συνθετικά μέσα. Για παράδειγμα, μια ψηφιακή φωτογραφική μηχανή ή ένας scanner μεταφέρει αυτόματα την εικόνα ενός αντικειμένου σε ψηφιακή μορφή κατάλληλη για χρήση στον υπολογιστή. Δηλαδή οι εικόνες είναι captured media. Το κείμενο, όταν αυτό πληκτρολογείται στον υπολογιστή είναι προφανώς συνθετικό μέσο. Αν όμως λαμβάνεται μέσω scanner και προγράμματος OCR πρέπει να θεωρηθεί ως captured.

· *Discrete versus continuous media*

Όταν ένας τύπος πληροφορίας έχει μόνο χωρική διάσταση ονομάζεται διακριτός. Αν υπάρχει και η συνιστώσα του χρόνου ονομάζεται συνεχής. Για παράδειγμα, οι εικόνες, το κείμενο και τα γραφικά είναι διακριτά, ενώ το βίντεο, ο ήχος και το animation είναι συνεχής.



Σχήμα 1 Ταξινόμηση ειδών πληροφορίας

Όλα αυτά τα μέσα που έχουν αναφερθεί ως τώρα απευθύνονται σχεδόν αποκλειστικά στην όραση και στην ακοή του ανθρώπου. Ένα σύστημα πολυμέσων δεν περιέχει απαραίτητα πληροφορίες για παραπάνω από μια αισθήσεις, παρόλο που κάτι τέτοιο είναι γενικά επιθυμητό.

1.4 Χαρακτηριστικά

Με βάση τον παραπάνω ορισμό, προκύπτουν τέσσερα χαρακτηριστικά για τα συστήματα πολυμέσων που μας ενδιαφέρουν:

- *Πρέπει να ελέγχονται από υπολογιστή.*

Δηλαδή η παρουσίαση της πληροφορίας γίνεται μέσω του υπολογιστή και ελέγχεται από αυτόν.

- *Είναι ολοκληρωμένα (integrated).*

Η ολοκλήρωση υπονοεί ότι ο αριθμός των υποσυστημάτων είναι κατά το δυνατόν ελάχιστος και ενσωματωμένος στον υπολογιστή. Παράδειγμα ολοκλήρωσης αποτελεί ή οθόνη του υπολογιστή που χρησιμοποιείται για την απεικόνιση κειμένου, εικόνας και βίντεο.

- *Η πληροφορία πρέπει να είναι σε ψηφιακή μορφή.*

Το χαρακτηριστικό αυτό είναι απόρροια της απαίτησης για έλεγχο και παρουσίαση μέσω υπολογιστή. Το πως γίνεται η μεταφορά κάθε τύπου πληροφορίας σε ψηφιακή μορφή, καθώς και τα πλεονεκτήματα της ψηφιακής αναπαράστασης της πληροφορίας θα εξεταστούν στο επόμενο κεφάλαιο.

- *Το interface με το χρήστη πρέπει να επιτρέπει αλληλεπίδραση (interaction).*

Αν και δεν περιλαμβάνεται ευθέως στον ορισμό, η δυνατότητα αυτή επιτρέπει την δημιουργία εφαρμογών με περισσότερες δυνατότητες από την απλή παρουσίαση της πληροφορίας (όπως γίνεται για παράδειγμα μέσω ενός video-player ή ενός CD-player) και είναι ιδιαίτερο χαρακτηριστικό των ελεγχόμενων μέσω υπολογιστή πολυμέσων.

1.5 Αυτόνομα και δικτυωμένα πολυμέσα

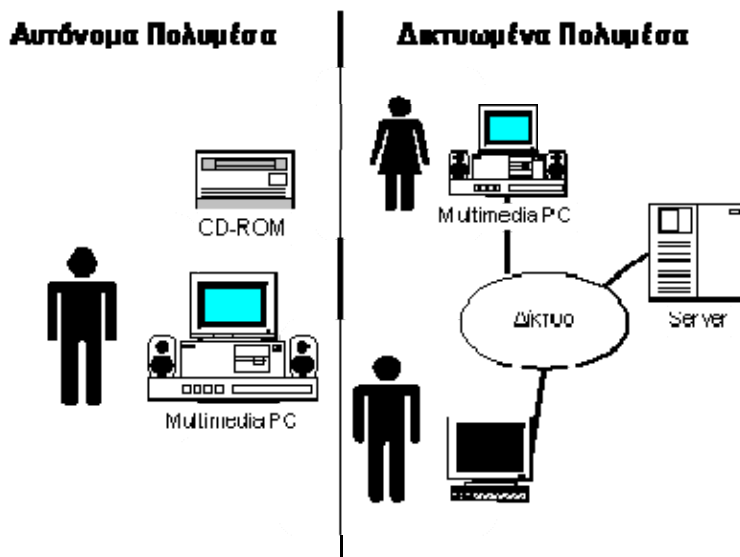
Ο όρος αυτόνομα ή τοπικά πολυμέσα αναφέρεται σε εφαρμογές που χρησιμοποιούν μόνο τον υπολογιστή στον οποίο τρέχουν. Κατά συνέπεια, ο υπολογιστής αυτός πρέπει να έχει όλες τις απαραίτητες υπομονάδες όπως:

- επεξεργαστή (όχι τερματικό δηλαδή)
- ικανό υποσύστημα γραφικών και ήχου
- ηχεία, μικρόφωνο

- αρκετά αποθηκευτικά μέσα
- κάποιας μορφής οπτικό δίσκο συνήθως CD-ROM

Πολλές όμως φορές είναι επιθυμητό οι εφαρμογές πολυμέσων να επικοινωνούν μέσω δικτύου με άλλους υπολογιστές για δύο λόγους:

- Την υποστήριξη εφαρμογών οι οποίες είναι εγγενώς δικτυακές. Παραδείγματα τέτοιων εφαρμογών είναι το ηλεκτρονικό ταχυδρομείο πολυμέσων και η τηλεδιάσκεψη.
- Την υλοποίηση του μοντέλου πελάτη-εξυπηρετητή(**client-server**). Πολλές φορές αν και μια εφαρμογή πολυμέσων μπορεί κάλλιστα να υλοποιηθεί σε έναν υπολογιστή μόνο, για λόγους οικονομίας του υλικού, είναι επιθυμητό να μπορεί να αξιοποιεί και υποσυστήματα που ανήκουν σε άλλους υπολογιστές. Χαρακτηριστική περίπτωση είναι η ύπαρξη ενός υπολογιστή με μεγάλα αποθηκευτικά μέσα (εξυπηρετητής) προσπελάσιμα μέσω δικτύου και από άλλους υπολογιστές με περιορισμένες δυνατότητες αποθήκευσης (πελάτες).



Σχήμα 2. Αυτόνομα και δικτυωμένα πολυμέσα

1.6 Φιλοσοφία των Hypertext και Hypermedia

Hyper-text. Το πρόθεμα σημαίνει “υπέρ” και χρησιμοποιήθηκε ευρύτατα στις αρχές του αιώνα κυρίως στα πεδία της φυσικής και των μαθηματικών για να περιγράψει ένα παράξενο, νέο είδος “χώρου” που ορίστηκε από τη θεωρία της σχετικότητας του Einstein-“υπερχώρος”. Ο χώρος αυτός του Einstein είναι ένας χώρος παρατηρημένος από μία άλλη οπτική γωνία, ένα καινούργιο είδος χώρου. Το ίδιο συμβαίνει και με το hypertext. Είναι κείμενο παρατηρημένο από μία άλλη οπτική γωνία, ένα νέο είδος κειμένου.

Πολύ απλά, μπορούμε να πούμε ότι το hypertext είναι μη γραμμικό ή αλλιώς **μη σειριακό κείμενο**. Αυτό σημαίνει ότι το κείμενο είναι έτσι οργανωμένο ώστε να μπορούμε εύκολα να περάσουμε από ένα θέμα σε κάποιο άλλο χωρίς να είναι απαραίτητο να διαβάσουμε όλο το κείμενο με μία σταθερή και προκαθορισμένη σειρά. Αν και η έννοια του hypertext είναι στενά συνδεδεμένη με τον χώρο της πληροφορικής και των υπολογιστών, εν τούτοις μπορούμε να βρούμε την έννοια του hypertext σε απλά και καθημερινά έγγραφα, όπως οι εφημερίδες και τα λεξικά. Η εφημερίδα λοιπόν είναι ένα παράδειγμα hypertext εγγράφου. Γιατί; Γιατί διαβάζουμε όποια άρθρα θέλουμε παραλείποντας αυτά που δε μας ενδιαφέρουν. Δεν χρειάζεται να διαβάσουμε όλη την εφημερίδα από την αρχή προς το τέλος, αλλά είτε ψάχνουμε μέχρι να βρούμε κάτι που να μας ενδιαφέρει είτε βλέπουμε στη πρώτη σελίδα τι μας ενδιαφέρει και πάμε στη σελίδα που βρίσκεται αυτό για να το διαβάσουμε λεπτομερώς. Αντίθετα ένα μυθιστόρημα δε μπορεί ασφαλώς να χαρακτηριστεί ως hypertext έγγραφο, αφού θα πρέπει να το διαβάσουμε σελίδα προς σελίδα χωρίς να μπορούμε να παραλείψουμε οποιοδήποτε κομμάτι αυτού και πολύ περισσότερο χωρίς να μπορούμε να βρούμε τα καλύτερα τμήματά του.

Ένας άλλος ορισμός που θα μπορούσαμε να δώσουμε για την έννοια του hypertext είναι ο παρακάτω :

“Hypertext είναι μία νέα προσέγγιση στη διαχείριση πληροφορίας, όπου οι πληροφορίες αποθηκεύονται σε ένα δίκτυο κόμβων οι οποίοι συνδέονται μεταξύ τους με συνδέσμους.”

Στον παραπάνω ορισμό χρησιμοποιήσαμε τον όρο *πληροφορία* και όχι *κείμενο* για να γενικεύσουμε τον όρο. Παρατηρούμε, λοιπόν, ότι οι κόμβοι μπορούν να περιέχουν εκτός από κείμενο, γραφικά, εικόνες, ήχο, video ή οποιοδήποτε άλλο είδος πληροφορίας. Έτσι, μπορούμε να πούμε, ότι όταν έχουμε να κάνουμε με κείμενο, τότε μιλάμε για hypertext, ενώ όταν το βάρος πέφτει σε άλλες μορφές πληροφοριών της τεχνολογίας πολυμέσων (**multimedia**), τότε μιλάμε για **hypermedia**. Είναι ενδεικτικό δε, ότι στη βιβλιογραφία, σχεδόν πάντα τα συστήματα hypertext και **hypermedia** ορίζονται μαζί, ενώ δεν είναι λίγοι αυτοί που χρησιμοποιούν τους δύο αυτούς όρους εντελώς ισοδύναμα. Παρακάτω, δείχνουμε σχηματικά πως συνδέονται και πόσο πολύ συνυφασμένα είναι τα **hypertext** και **hypermedia συστήματα**.

Αναφέραμε προηγουμένως ότι ένα σύστημα hypertext είναι μη γραμμικό. Αυτό είναι ίσως κάτι που θα έπρεπε να αναλύσουμε λίγο παραπάνω γιατί ο όρος “**μη γραμμικό**” μπορεί να μας παραπλανήσει. Φυσικά, και κάθε ξεχωριστό μονοπάτι μέσα σε ένα hypertext σύστημα είναι **γραμμικό**: ο αναγνώστης εξακολουθεί να διαβάζει ή να βλέπει ή να ακούει διάφορα “**αντικείμενα**” με κάποια αλληλουχία, η οποία θα είναι γραμμική υπό την έννοια ότι κάθε τι θα ακολουθείται από κάτι άλλο. Αυτό που κάνει το hypertext να είναι **μη γραμμικό** είναι η *δυνατότητα επιλογής*, δηλαδή η αλληλεπίδραση του χρήστη ο οποίος αποφασίζει σε κάθε χρονική στιγμή ποιο μονοπάτι θα ακολουθήσει ανάμεσα από μία πλειάδα διαφορετικών μονοπατιών-επιλογών. Υπό αυτή την έννοια το hypertext απεικονίζει πολύ καλά την “πραγματική ζωή” και απευθύνει συνεχώς στον αναγνώστη την ίδια ερώτηση : “Τι θέλεις να διαβάσεις μετά; “. Φυσικά, διαφορετικοί αναγνώστες

ανταποκρίνονται με διαφορετικό τρόπο σε αυτήν την ερώτηση, ώστε ο καθένας να ορίζει το δικό του μονοπάτι μέσα στο hypertext σύστημα. Στην περίπτωση, μάλιστα, των hypermedia συστημάτων, η παραπάνω ερώτηση διαφοροποιείται και επαναδιατυπώνεται ως εξής: «Τι θέλεις να διαβάσεις ή να ακούσεις ή να δεις μετά;». Για παράδειγμα θα μπορούσε να διατυπωθεί η εξής ερώτηση : «Θέλεις να διαβάσεις το τάδε ποίημα, θέλεις να το ακούσεις ή θέλεις να δεις τον ποιητή να απαγγέλλει το ποίημά του;». Το hypertext, λοιπόν, ενθαρρύνει την εξερεύνηση από την πλευρά του αναγνώστη.

1.7 Ιστορική αναδρομή-Υπάρχοντα συστήματα

Μπορεί οι έννοιες των hypertext και hypermedia να έγιναν γνωστές στο τέλος της δεκαετίας του '80, αλλά έχουν μία ιστορία τουλάχιστον 30 ετών με αυτούς τους όρους, ενώ για πρώτη φορά η έννοια του hypertext σκιαγραφήθηκε το 1945 από τον Vannevar Bush. Ο Bush ήταν ο πρώτος που παρατήρησε ότι οι υπάρχοντες μέθοδοι για την διαχείριση και αναπαράσταση των πληροφοριών το 1945 ήταν ανεπαρκείς. Ταυτόχρονα, έβλεπε ότι η ανακάλυψη του τρανζίστορ άνοιγε το δρόμο για πολύπλοκες μηχανές μεγάλης αξιοπιστίας. Έτσι λοιπόν ο Bush οραματίστηκε μία ιδιαίτερα ισχυρή μηχανή που θα μπορούσε να αποθηκεύσει κάθε είδους πληροφορία: βιβλία, γράμματα, σημειώσεις, σχόλια, σχήματα, φωτογραφίες κλπ, οργανωμένα σε κάποιο είδος εγγράφων. Την μηχανή αυτή την ονόμασε *Memex* και ένα σημαντικότερο χαρακτηριστικό της ήταν η δυνατότητα διασύνδεσης μεταξύ δύο οποιονδήποτε σημείων των εγγράφων του συστήματος. Αυτό ακριβώς το χαρακτηριστικό ήταν που έδινε στο χρήστη τη δυνατότητα να πάρει από το σύστημα πληροφορίες με ένα τρόπο συσχετιστικό - τρόπο με τον οποίο δουλεύει και το ανθρώπινο μυαλό πηδώντας από τη μία σκέψη στην άλλη. Βέβαια όλα αυτά τα ιδιαίτερα πρωτοποριακά και φιλόδοξα σχέδια, για εκείνη την εποχή ήταν απλώς ένα όνειρο, μία ιδεατή μηχανή.

Δεκατέσσερα χρόνια μετά τις πρώτες αυτές ιδέες του Bush, ένας άλλος ερευνητής, ο Doug Engelbart που προβληματιζόταν για την αλληλεπίδραση ανθρώπου - μηχανής, διαβάζοντας το άρθρο του Bush αποφάσισε να ασχοληθεί με την υλοποίηση ενός hypertext συστήματος. Έτσι λοιπόν γι' αυτό το σκοπό, το 1959 ξεκίνησε ένα ερευνητικό project στο ινστιτούτο ερευνών του Stanford (Stanford Research Institute) στο οποίο εργαζόταν ο Engelbart με το όνομα «Augmenting Human Intellect». Το 1963 ο Engelbart δημοσίευσε τις ιδέες του υπό τον τίτλο : «A Conceptual Framework for the Augmentation of Man's Intellect» οι οποίες απέφεραν πέντε χρόνια αργότερα το πρώτο hypertext σύστημα, γνωστό σαν *NLS (oN Line System)* ή όπως καθιερώθηκε αργότερα σαν «*Augment*».

Έτσι, το 1968 ο Engelbart παρουσίασε σε μία διάσκεψη επιστημόνων πληροφορικής, το πρώτο hypertext σύστημα, που σε αυτή τη πρώτη φάση αποτελούνταν από αρχεία που ήταν οργανωμένα σε τμήματα των 3000 λέξεων το πολύ, που μπορούσαν να συνδέονται μεταξύ τους, είτε σχηματίζοντας κάποια ιεραρχία, είτε όχι. Το σύστημα αυτό με τα χρόνια αναπτύχθηκε κι άλλο, κι έγινε γνωστό όπως είπαμε με το όνομα "Augment" και σήμερα χρησιμοποιείται για να παρέχει επικοινωνία μεταξύ μηχανικών λογισμικού. Τέλος είναι σημαντικό να παρατηρήσουμε ότι ο Engelbart ασχολούμενος με την ανάπτυξη του συστήματος αυτού και για την διευκόλυνση των χρηστών του, επινόησε το ποντίκι που

έμελλε να αποτελέσει στα επόμενα χρόνια βασικό εξοπλισμό για όλα τα hypertext/hypermedia συστήματα και όχι μόνο.

Ένας ακόμα σημαντικός σταθμός στην εξέλιξη της έννοιας του hypertext αλλά και των hypertext συστημάτων ήταν η άποψη και οι ιδέες του Ted Nelson που δημοσιοποιήθηκαν με τη τρίτη και τελική τους μορφή το 1965. Αξίζει εδώ να σημειωθεί ότι ο Nelson ήταν αυτός που χρησιμοποίησε πρώτος τον όρο hypertext. Οι ιδέες αυτές εντάσσονταν σε κάποιο project που ξεκίνησε ο Nelson το 1960 στα πλαίσια κάποιου μεταπτυχιακού μαθήματος που είχε πάρει στο πανεπιστήμιο του Harvard. Το project αυτό γνωστό ως "*Xanadu Project*" πήρε το όνομα του από το μυστήριο παλάτι που αναφέρεται στο ποίημα "Kubla Khan" του S.T. Coleridge και είχε ως στόχο του, την ενοποίηση και ενσωμάτωση της παγκόσμιας λογοτεχνίας σε ένα σύστημα άμεσης επικοινωνίας, και την χρήση του hypertext για την διασύνδεση διαφόρων τμημάτων της, με τρόπο τέτοιο ώστε να διευκολύνεται ο χρήστης.

Στο τέλος της δεκαετίας του '60 ο Nelson προσκλήθηκε από το πανεπιστήμιο του Brown όπου και παρέμεινε για αρκετά χρόνια βοηθώντας σε κάποιο project που είχε σχέση με την ανάπτυξη ενός hypertext editing συστήματος (FRES). Στα επόμενα χρόνια ο Nelson συνέχισε να εργάζεται στο hypertext σύστημα που είχε αναπτυχθεί με το Xanadu Project, παρά την έλλειψη επαρκών πόρων. Το 1988 η εταιρεία Autodesk, μία μεγάλη εταιρεία ανάπτυξης λογισμικού γνωστή από το διάσημο AutoCad, ανέλαβε τα δικαιώματα του Xanadu Project, με αποτέλεσμα να προκύψει ένα εμπορικό hypertext σύστημα που λειτουργεί σε τοπικό δίκτυο, με εξυπηρετητή ένα σταθμό εργασίας Sun.

Σήμερα, η πιο διαδεδομένη και μεγάλη σε μέγεθος εφαρμογή των hypertext-hypermedia είναι το **World Wide Web (WWW)** στο Internet. Το WWW είναι μία οικογένεια πρωτοκόλλων και συστημάτων που έχουν ως στόχο τη διασύνδεση διαφόρων ειδών πληροφορίας σύμφωνα με την ιδέα των hypermedia. Ξεκίνησε το 1992 ως πείραμα στο CERN στη Γενεύη και έχει εξελιχθεί σε μία παγκόσμια προσπάθεια που έχουν ενστερνιστεί όλες οι εταιρείες πληροφορικής. Με βάση τις συμβάσεις και τα πρωτόκολλα του WWW, δημιουργείται ένας χώρος δομημένης ψηφιακής πληροφορίας πολλών ειδών, που ονομάζεται **hyperspace**.

Αυτοί μπορούμε να πούμε ότι ήταν οι κυριότεροι σταθμοί στην εξελικτική πορεία του hypertext. Από την επίδειξη του Engelbart, το 1968 και μετά έχουν αναπτυχθεί πολυάριθμα hypertext και hypermedia συστήματα, πειραματικά ή εμπορικά για εξειδικευμένες εφαρμογές ή για γενική χρήση, αρχικά σε μεγάλα υπολογιστικά συστήματα και αργότερα - με την ανακάλυψη τους - και σε μίνι υπολογιστές (PC compatibles ή Macintosh). Θα αναφέρουμε μερικά ενδεικτικά παραδείγματα τέτοιων συστημάτων, έτσι ώστε να δούμε την εξέλιξη τους αλλά και τις δυνατότητες και τους περιορισμούς που υπάρχουν σε αυτά.

FRES (File Retrieval and Editing System) : Σύστημα hypertext που αναπτύχθηκε και λειτούργησε σε ακαδημαϊκό περιβάλλον, στο Πανεπιστήμιο Brown, από το 1960. Επιτρέπει να οριστούν λέξεις-κλειδιά πάνω στα διαθέσιμα έγγραφα και να δημιουργηθούν σύνδεσμοι μεταξύ εγγράφων.

Neptune : Σύστημα hypertext που δημιουργήθηκε στην εταιρεία Tektronix το 1986 από τους N.Delisle και M.Schwartz. Αν και περιέχει αρκετά ενδιαφέροντα χαρακτηριστικά, δεν έχει γνωρίσει ιδιαίτερη δημοσιότητα. Βασίζεται σε ένα hypertext μοντέλο που ονομάζεται Hypertext Abstract Machine (**HAM**) το οποίο παρέχει λειτουργίες δημιουργίας, μετατροπών και προσπέλασης κόμβων και συνδέσμων, ενώ περιέχει και αρκετά χαρακτηριστικά που είναι χρήσιμα στις κατανεμημένες βάσεις δεδομένων. Το σύστημα Neptune λειτουργεί σε περιβάλλον δικτύου, υποστηρίζει πολλούς χρήστες και χρησιμοποιεί δοσοληψίες (**transaction management**) για να διατηρεί την ακεραιότητα και τη συνέπεια των δεδομένων.

BALSA (Brown Algorithm Simulator and Animator) : Περιβάλλον με χαρακτηριστικά συστήματος hypermedia που υποστηρίζει την επεξεργασία κειμένου και γραφικών και επιτρέπει τη δημιουργία συνδέσμων και τον ορισμό φίλτρων για τις διαθέσιμες πληροφορίες, τα οποία λειτουργούν όπως οι όψεις (**views**) στις σχεσιακές βάσεις δεδομένων.

EDS (Electronic Document System) : Σύστημα hypermedia που δημιουργήθηκε στο Πανεπιστήμιο Brown. Η σχεδίαση του στηρίχθηκε στα συμπεράσματα από το FRES και η υλοποίηση του ολοκληρώθηκε το 1982.

Intermedia : Σύστημα hypermedia που αναπτύχθηκε επίσης στο Πανεπιστήμιο Brown και συνεπώς διαθέτει όλα τα χαρακτηριστικά των προγενέστερων συστημάτων που αναπτύχθηκαν στο ίδιο πανεπιστήμιο, FRES και EDS. Παρέχει ένα σύνολο εργαλείων για δεδομένα διαφορετικής μορφής που συνεργάζονται μεταξύ τους, ενώ μπορεί να αποθηκεύσει κείμενο, γραφικά και δεδομένα από άλλου είδους εφαρμογές, να δημιουργήσει συνδέσμους και λέξεις- κλειδιά τόσο για τα έγγραφα όσο και για τους συνδέσμους του συστήματος. Πολύ σημαντικό και ενδιαφέρον χαρακτηριστικό του συστήματος αποτελεί η δυνατότητα ορισμού όψεων (**perspectives**) ανάλογων με αυτών που υπάρχουν στις σχεσιακές βάσεις δεδομένων. Ακόμα υπάρχουν τοπικοί αλλά και γενικοί χάρτες του hypermedia δικτύου που βοηθούν στο προσανατολισμό των χρηστών μέσα στο σύστημα. Τέλος, το Intermedia κατασκευάστηκε σε λειτουργικό σύστημα UNIX και τρέχει σε IBM RT/PC και SUN σταθμούς εργασίας.

NoteCards : Σύστημα hypermedia που αναπτύχθηκε στο Intelligent Systems Lab της Xerox PARC από τους Halasz, Moran και Trigg και ολοκληρώθηκε το 1987. Υποστηρίζει κείμενο γραφικά και εικόνες που αποθηκεύονται σε κάρτες που ονομάζονται notecards, μεταξύ των οποίων μπορούν να δημιουργηθούν σύνδεσμοι. Οι χρήστες έχουν τη δυνατότητα όχι μόνο να δουν τη δομή του δικτύου hypertext αλλά και να την αλλάξουν με έναν ειδικό editor. Τέλος, υποστηρίζεται η περιπλάνηση και σε μικρότερο βαθμό η αναζήτηση κειμένου.

KMS (Knowledge Management System) : Εμπορικό σύστημα hypermedia που υποστηρίζει κείμενο, γραφικά και ψηφιοποιημένες εικόνες. Ολοκληρώθηκε το 1987 και αποτελεί την εμπορική εκδοχή του προγενέστερου συστήματος ZOG και καρπούς προσπαθειών αρκετών χρόνων. Η όλη προσπάθεια ανάπτυξης ενός hypermedia συστήματος ξεκίνησε το 1972 στο Πανεπιστήμιο Carnegie-Mellon (**CMU**) με το ZOG project που τελικά τέλειωσε το 1984 και απέφερε το σύστημα ZOG. Η ανάγκη όμως για

μια εμπορική έκδοση του ίδιου συστήματος ώθησε την Knowledge Systems πλέον στην υλοποίηση του KMS. Όσον αφορά τις δυνατότητες του KMS, είναι σχεδιασμένο για τη δημιουργία βάσεων δεδομένων σε περιβάλλοντα μεγάλων οργανισμών και εταιρειών. Οι πληροφορίες αποθηκεύονται σε πλαίσια (frames) ενώ μπορούν να δημιουργηθούν σύνδεσμοι μεταξύ πλαισίων και να σχηματιστούν ιεραρχίες από πλαίσια. Οι σύνδεσμοι αυτοί μπορούν να δείχνουν σε εξωτερικά προγράμματα που θα ενεργοποιούνται όταν θα διασχίζονται οι αντίστοιχοι σύνδεσμοι. Επίσης υπάρχει η δυνατότητα αναζήτησης κειμένου στην ιεραρχία των πλαισίων, καθώς και δυνατότητα για ακύρωση ενεργειών και οπισθοχώρησης, ενώ δεν παρέχεται γραφική εποπτεία του hypermedia δικτύου.

Guide : Σύστημα hypertext που αναπτύχθηκε στο Πανεπιστήμιο του Kent στη Μεγάλη Βρετανία και διατίθεται σε μίνι υπολογιστές από την εταιρεία Owl International. Παρέχει έναν ιδιαίτερα ευέλικτο τρόπο κατασκευής ηλεκτρονικών εγγράφων και μία ποικιλία τρόπων σύνδεσης αυτών. Υποστηρίζει κείμενο και γραφικά που αποθηκεύονται σε κόμβους οι οποίοι ενώνονται με συνδέσμους. Οι σύνδεσμοι αυτοί μπορούν να είναι διαφόρων τύπων και μπορούν να σχηματίζουν ιεραρχίες κόμβων ή να καταλήγουν σε εξωτερικά εκτελέσιμα προγράμματα.

HyperTIES : Σύστημα που σχεδιάστηκε στο Πανεπιστήμιο του Maryland και ολοκληρώθηκε το 1987. Ουσιαστικά πρόκειται για μία ηλεκτρονική εγκυκλοπαίδεια. Το συνθετικό TIES άλλωστε του hyperTIES προέρχεται από το "The Interactive Encyclopedia System". Υποστηρίζει κείμενο και γραφικά, επιτρέπει τη δημιουργία συνδέσμων και διαθέτει μηχανισμούς περιπλάνησης και αναζήτησης για την ανάκτηση πληροφοριών.

TextVision : Σύστημα hypertext που δημιουργήθηκε στο Πανεπιστήμιο του Twente στην Ολλανδία. Δίνει ιδιαίτερο βάρος στη σημασιολογική σύνδεση των πληροφοριών σε ένα γράφο και προορίζεται για εκπαιδευτικές εφαρμογές. Στο σύστημα αυτό συναντάμε για πρώτη φορά την έννοια της κεντρικότητας των κόμβων (node centrality) που υποδεικνύει πόσο σημαντικός είναι ένας κόμβος, με βάση το πλήθος των συνδέσμων που δείχνουν σ' αυτόν. Επίσης δίνεται η δυνατότητα στον χρήστη να αλλάζει εύκολα τη δομή του δικτύου hypertext.

HyperNet : Σύστημα hypertext με πολλές ομοιότητες με το σύστημα TextVision. Παρέχει ισχυρά εποπτικά μέσα και εργαλεία για το χειρισμό της δομής του δικτύου hypertext. Επίσης υποστηρίζει την περιπλάνηση ως τη βασική διαδικασία ανάκτησης πληροφοριών.

HyperCard : Εμπορικό σύστημα hypertext που αναπτύχθηκε από την Apple για υπολογιστές Macintosh. Οι κόμβοι στο σύστημα αυτό ονομάζονται κάρτες, ενώ οι σχετιζόμενες κάρτες σχηματίζουν τις λεγόμενες στοίβες μέσα στις οποίες μπορούν οι χρήστες να περιπλανηθούν. Το hypercard διαθέτει επίσης μία δική του γλώσσα χειρισμού των δεδομένων, την HyperTalk με την οποία μπορούμε να γράφουμε scripts που καθορίζουν την συμπεριφορά των διαφόρων αντικειμένων αλλά και επιτρέπουν τη σύνδεση των καρτών. Οι σύνδεσμοι δεν υποστηρίζονται άμεσα αλλά προσομοιώνονται με κουμπιά (**buttons**) που δημιουργούμε μέσα στις κάρτες, των οποίων τα scripts περιέχουν απλώς εντολές ανάκτησης συγκεκριμένων καρτών. Οι τελευταίες εκδόσεις του hypercard επιτρέπουν επίσης σύνδεση με video και CD-ROMs, γεγονός που το αναβαθμίζει σε

hypermedia σύστημα. Τέλος πρέπει να πούμε ότι το hypercard δεν είναι το μόνο hypertext σύστημα για υπολογιστές Macintosh, αλλά έχουν κυκλοφορήσει και άλλα παρόμοια όπως το SuperCard και το HyperGate.

HyperPad : Σύστημα hypertext με μεγάλες ομοιότητες με το hypercard και με δικιά του γλώσσα χειρισμού δεδομένων, την PADtalk. Σημαντικός περιορισμός του συστήματος αυτού είναι ότι υποστηρίζει μόνο γραφικά χαρακτήρων. Υπάρχει επίσης ένα πρόγραμμα μετατροπής στοιβών του hypercard σε hyperPad.

Hyperdoc : Αρκετά ισχυρό εργαλείο ανάπτυξης hypertext/hypermedia εφαρμογών που δημιουργήθηκε στη Γαλλία. Παρέχει έναν editor γραφικών καθώς και μία δική του γλώσσα χειρισμού δεδομένων. Επίσης υποστηρίζονται αρκετοί τύποι εξωτερικών αρχείων (π.χ. dBase) καθώς και σύνδεση με βιντεοδίσκο.

IBM LinkWay : Ευέλικτο σύστημα hypertext/hypermedia που υποστηρίζει κείμενο, γραφικά, εικόνες, μουσική, αφήγηση και full-motion video και δυνατότητα περιπλάνησης στα σχετιζόμενα τμήματα πληροφορίας που συνδέονται μεταξύ τους. Παρέχει επίσης editor κειμένου, editor εικόνων καθώς και ενσωματωμένη γλώσσα χειρισμού δεδομένων.

Ize : Σύστημα hypertext διαχείρισης κειμένου. Οι κόμβοι ονομάζονται texts, περιέχουν μόνο κείμενο που μπορεί να είναι από μία λέξη μέχρι μεγέθους 30K. Σε ένα δίκτυο hypertext που ονομάζεται textbase μπορούν να υπάρχουν μέχρι 32000 texts. Σε κάθε κείμενο εκχωρείται τουλάχιστον μία λέξη-κλειδί με βάση τις οποίες γίνεται η ανάκτηση των κειμένων.

HyperSprint : Hypertext interface για τον επεξεργαστή κειμένου Sprint που διατίθεται από την Borland. Ενσωματώνει όλες τις δυνατότητες του αναφερόμενου επεξεργαστή κειμένου, μαζί με hypertext χαρακτηριστικά όπως οι συνδέσεις μεταξύ δύο σημείων του ίδιου αρχείου ή συνδέσεις μεταξύ διαφορετικών αρχείων.

Multimedia ToolBook : Hypertext/Hypermedia σύστημα, σχεδόν όμοιο με το HyperCard. Υποστηρίζει κείμενο, γραφικά, ψηφιοποιημένες εικόνες, ήχο, full-motion video, καθώς και σύνδεση με CD-ROM και βιντεοδίσκο. Διαθέτει επίσης μία αρκετά ισχυρή ενσωματωμένη γλώσσα χειρισμού δεδομένων που μπορεί όμως να επεκταθεί ακόμα περισσότερο. Όσον αφορά την ορολογία που χρησιμοποιεί το ToolBook, οι κόμβοι στους οποίους αποθηκεύονται οι πληροφορίες ονομάζονται σελίδες (**pages**) (ανάλογο του card στο HyperCard), ενώ ο αντίστοιχος όρος της στοιβας, εδώ είναι το βιβλίο (**book**).

Έχοντας λοιπόν ορίσει τις έννοιες hypertext και hypermedia, έχοντας δει τις πρώτες προσπάθειες που έγιναν για την ανάπτυξη σχετικών συστημάτων και βλέποντας και πολλά από τα υπάρχοντα συστήματα, μπορούμε να προχωρήσουμε και να δούμε τα βασικά στοιχεία αυτής της τεχνολογίας.

1.8 Βασικά δομικά στοιχεία

Όπως είναι φανερό από αυτά που έχουν αναφερθεί μέχρι τώρα, το hypertext είναι μία νέα προσέγγιση στη διαχείριση πληροφορίας, όπου οι πληροφορίες αποθηκεύονται σε ένα δίκτυο κόμβων που συνδέονται μεταξύ τους με συνδέσμους. Από τη παραπάνω περιγραφή προκύπτει

ότι τα δομικά στοιχεία του hypertext είναι οι *κόμβοι (nodes)* και οι *σύνδεσμοι* μεταξύ τους (*links*).

1.8.1 Κόμβοι

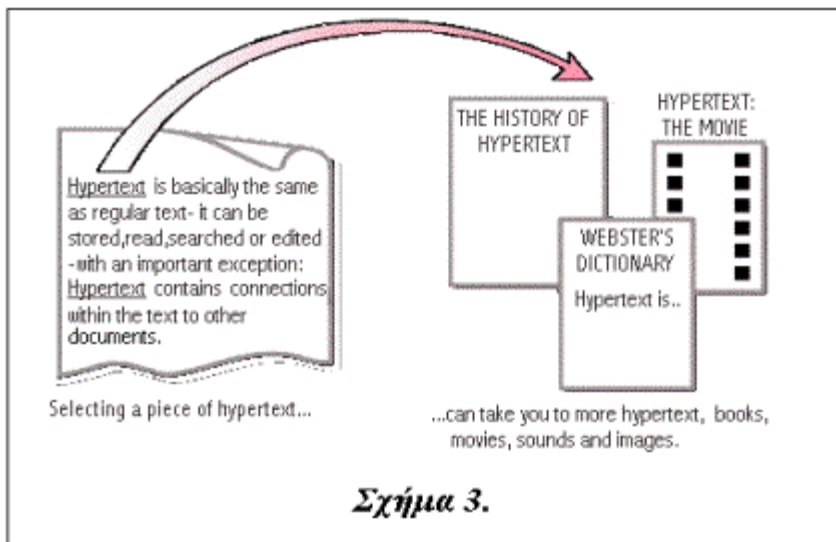
Ως κόμβος ορίζεται μία συλλογή πληροφοριών σχετικών με κάποιο αντικείμενο. Σε διαφορετικά hypertext συστήματα, χρησιμοποιούνται διαφορετικοί όροι για την έννοια του κόμβου. Οι κόμβοι είναι ουσιαστικά μονάδες πληροφορίας οι οποίες γενικά είναι αυτάρκειες. Εξαιτίας, της αρθρωτής φύσης του hypertext είναι γενικά ωφέλιμο να υπάρχει μία μορφή πλεονάζουσας πληροφορίας στους κόμβους έτσι ώστε κάθε υπομονάδα να είναι κατανοητή

1.8.2 Σύνδεσμοι

Όπως αναφέραμε και παραπάνω οι κόμβοι συνδέονται μεταξύ τους με τους συνδέσμους (*links*), γεγονός που επιτρέπει στο χρήστη να κινείται από τον ένα κόμβο στον άλλο. Άλλωστε στα περισσότερα υπάρχοντα hypertext και hypermedia συστήματα, οι σύνδεσμοι υλοποιούνται σαν λειτουργίες "μεταπήδησης" (*jump operations*) από τον ένα κόμβο στον άλλο, χωρίς να φέρουν κάποια σημασιολογική πληροφορία. Μία ενδιαφέρουσα σημασιολογική προσέγγιση για τους συνδέσμους είναι ότι αποτελούν πληροφορία εμφυτευμένη σε ένα κόμβο ώστε να τον συνδέει με κάποιο συγκεκριμένο τρόπο με κάποιον άλλο κόμβο. Σε αυτή τη περίπτωση λέμε ότι οι σύνδεσμοι είναι κάποιου συγκεκριμένου τύπου (*typed (or named) links*). Π.χ. μπορούμε να έχουμε συνδέσμους αιτίου-αποτελέσματος, θέσης-αντίθεσης ή γενικού-ειδικού περιεχομένου. Τελειώνοντας αυτή τη παράγραφο θα δώσουμε τρεις ακόμα ορισμούς σχετικά με τους κόμβους και τους συνδέσμους.

Το τμήμα του περιεχομένου ενός κόμβου απ' όπου ξεκινά ένας σύνδεσμος μπορεί να ονομαστεί *περιοχή αναχώρησης (departure region)* του συνδέσμου αυτού, και το τμήμα ενός κόμβου όπου καταλήγει ένας σύνδεσμος μπορεί να ονομαστεί *περιοχή άφιξης (arrival region)* του συνδέσμου αυτού. Η περιοχή αναχώρησης ή η περιοχή άφιξης ενός συνδέσμου λέγονται *άγκυρα (anchor)* του συνδέσμου αυτού και στις περισσότερες περιπτώσεις είναι μία λέξη ενός κειμένου ή στην περίπτωση του Web μπορεί να είναι "λέξη" άλλης μορφής (εικόνα, γραφικά). Στην ιδανική περίπτωση θα πρέπει να είναι προφανές από την συναφή έκφραση της άγκυρας που οδηγεί ένας σύνδεσμος και τι είδους πληροφορία παρέχει. Παρακάτω, παραθέτουμε δύο απλά σχηματικά παραδείγματα hypertext συστημάτων. Στο πρώτο, το hypertext σύστημα απεικονίζεται ως ένα δίκτυο κόμβων που συνδέονται μεταξύ τους με συνδέσμους.

Στο δεύτερο έχουμε μία πιο παραστατική απεικόνιση όπου εκτός από τους κόμβους και τους συνδέσμους φαίνονται και οι άγκυρες. Έτσι, παρατηρούμε ότι μπορούμε να έχουμε σύνδεσμο εντός ενός κόμβου ή να έχουμε σύνδεσμο που να συνδέει δύο διαφορετικούς κόμβους. Στη δεύτερη αυτή περίπτωση, ο σύνδεσμος μπορεί να ξεκινά από μία άγκυρα ενός κόμβου και να καταλήγει σε μία άγκυρα ενός άλλου κόμβου ή να καταλήγει σε ολόκληρο τον άλλο κόμβο. Τέλος, παραθέτουμε ένα ακόμη παράδειγμα hypertext συστήματος, όχι αφηρημένο, αλλά συγκεκριμένο. Από ένα κείμενο, μπορούμε πατώντας στη λέξη-κλειδί hypertext να μεταβούμε σε κάποιο άλλο σημείο όπου θα έχει περισσότερες πληροφορίες για ιστορικά στοιχεία, ερμηνεία του όρου ή ακόμα και κάποιο video.



1.8.3 Δομικά μοντέλα hypertext

Πριν αναφερθούμε στα διαφορετικά μοντέλα που υπάρχουν και που μπορούμε να χρησιμοποιήσουμε προκειμένου να δομήσουμε ένα hypertext σύστημα θα πρέπει να τονίσουμε ότι ο βασικός πυρήνας ενός hypertext συστήματος είναι ουσιαστικά δομή ανάλογη με αυτή ενός δένδρου αποφάσεων. Κάθε άγκυρα παριστά την δυνατότητα για μία επιλογή: να ακολουθήσει τον σύνδεσμο και έτσι να εγκαταλείψει το τρέχον περιεχόμενο και να αγνοήσει τον συγκεκριμένο σύνδεσμο. Αυτή η τεράστια δυνατότητα επιλογών πρέπει να προσεχθεί ιδιαίτερα, ιδίως όσον αφορά τους αρχάριους χρήστες γιατί μπορεί πολύ εύκολα να αποσπαστεί προσοχή τους και να χάσουν τον προσανατολισμό τους.

Υπάρχουν τέσσερις βασικοί τρόποι δόμησης ενός hypertext συστήματος, οι οποίοι είναι οι εξής :

- **Μη δομημένο hypertext** : κάθε κόμβος συνδέεται με κάθε άλλο κόμβο. Το σκεπτικό αυτής της προσέγγισης είναι ότι κάθε προσπάθεια δόμησης της πληροφορίας, όσο καλόπιστη κι αν είναι αυτή, μπορεί να αποτελεί παραβίαση με κάποιο τρόπο του συγκεκριμένου ενδιαφέροντος του αναγνώστη τη δεδομένη χρονική στιγμή. Το πρόβλημα με αυτή την προσέγγιση έγκειται στο ότι είναι εφικτή μόνο για πολύ μικρά hypertext συστήματα και τελικά η πολυπλοκότητά τους φαίνεται να είναι μεγαλύτερη από την ελευθερία κινήσεων που προσφέρουν.
- **Συμπερασματικό hypertext** : Η βασική ιδέα αυτού του τύπου δόμησης hypertext συστήματος είναι να μιμηθούμε τις γνώσεις του πιο έμπειρου και ειδικού πάνω στο συγκεκριμένο θέμα που εξετάζεται. Οι σύνδεσμοι και οι άγκυρες τοποθετούνται σύμφωνα με τις υποδείξεις του ειδικού. Αυτή η μέθοδος προσφέρει μία καλή καθοδήγηση μέσα στο σύστημα αλλά μπορεί να μην παίρνει υπόψη της την αδαημοσύνη ενός αρχάριου χρήστη στο συγκεκριμένο θέμα. Επίσης, ένα άλλο πρόβλημα αυτής της μεθόδου είναι η δυσκολία στο να βρούμε εκείνο τον ειδικό με τέτοια τεράστια και αλάθητη γνώση.

- **Επαγωγικό hypertext** : Αυτή η μέθοδος ξεκινά με ένα αδόμητο hypertext σύστημα και κατόπιν παρατηρεί τις κινήσεις των χρηστών μέσα στο σύστημα. Οι κινήσεις αυτές αναλύονται και κατόπιν οι πιο συνηθισμένες και διαδεδομένες απ' αυτές υπερθέτονται στο σύστημα. Αυτή η προσέγγιση είναι σίγουρα η πιο φιλική προς το χρήστη αλλά μπορεί να αποδειχθεί αρκετά δύσκολο και χρονοβόρο να βρούμε ποιες είναι πράγματι οι πιο δημοφιλείς αποφάσεις των χρηστών.

Οι δύο παραπάνω τρόποι δόμησης hypertext και hypermedia συστημάτων είναι αρκετά στατικοί και θεωρούν ότι το ολικό περιεχόμενο του συστήματος που φτιάχνουμε έχει δοθεί εκ των προτέρων. Ετσι, είναι αρκετά δύσκαμπτοι στην περίπτωση που θέλουμε να προσθέσουμε νέα θέματα στο σύστημά μας ή να επεκτείνουμε ήδη υπάρχοντα. Το κενό αυτό, έρχεται να καλύψει ο επόμενος τρόπος δόμησης hypertext και hypermedia συστημάτων, τον οποίο μπορούμε να χαρακτηρίσουμε :

- **Hypertext των σταδιακών μεταλλαγών** : Αυτή η μέθοδος αναπτύσσει κόμβους και συνδέσμους ακριβώς τη στιγμή που αυτοί χρειάζονται. Εν τέλει, όσον αφορά το Web, αυτή φαίνεται να είναι η πιο ρεαλιστική λύση καθότι όλα τα συστήματα διαρκώς αναπτύσσονται και αλλάζουν. Παρόλα αυτά και αυτή η προσέγγιση είναι αρκετά γενική και στερείται απαραίτητου φορμαλισμού.

1.8.4 Περιήγηση (navigation) και περιόδευση (browsing)

Η μετακίνηση από κόμβο σε κόμβο ακολουθώντας κάποιους συνδέσμους, αναφέρεται στη βιβλιογραφία ως **περιήγηση (navigation)** και ήταν ένας από τους λόγους που η τεχνολογία του hypermedia προωθήθηκε σαν μία νέα αντίληψη στο χώρο της ανάκτησης πληροφοριών. Ένας ακόμα όρος που αναφέρεται συχνά σαν συνώνυμο της περιπλάνησης είναι η **περιόδευση (browsing)**. Οι έννοιες όμως αυτές αν και έχουν κοντινό νοηματικά περιεχόμενο, διαφέρουν, και αυτή τη διαφορά θα προσπαθήσουμε να αναδείξουμε παρακάτω. Αυτό θα γίνει μέσα από τη μελέτη των εννοιών *ανάκτηση και εξερεύνηση πληροφορίας*. Ένας αρκετά ευρύς ορισμός για τον όρο **ανάκτηση πληροφοριών (information retrieval)** είναι αυτός που δίνεται από τους Salton και McGill στο βιβλίο τους "Introduction to modern information retrieval" (Εκδόσεις McGraw-Hill, 1983), σύμφωνα με τον οποίο : **"Η ανάκτηση πληροφοριών σχετίζεται με την αναπαράσταση, αποθήκευση, οργάνωση και προσπέλαση τμημάτων πληροφορίας."** Παρ' ότι ένας τέτοιος ορισμός κρίνεται επαρκής και ικανοποιητικός, η έννοια της ανάκτησης πληροφοριών τα τελευταία χρόνια έχει ταυτιστεί με αναζητήσεις δεικτοδοτούμενων βιβλιογραφικών βάσεων δεδομένων, βασιζόμενες σε κλειδιά και με χρησιμοποίηση boolean λογικής.

Παρ' όλα αυτά, η **εξερεύνηση πληροφορίας (information exploration)** είναι μία πολύ ευρύτερη διαδικασία που μπορεί να γίνει με πλήθος τρόπων και από διαφορετικούς ανθρώπους. Για παράδειγμα η περιδιάβαση των διαδρόμων μιας βιβλιοθήκης, είναι μια διαδικασία εξερεύνησης πληροφοριών αφού με αυτό το τρόπο μπορεί να βρούμε τυχαία κάτι που να μας ενδιαφέρει. Τι ακριβώς όμως εννοούμε με τον όρο εξερεύνηση πληροφορίας; Στο

"Αντιλεξικό της Νεοελληνικής Γλώσσας" του Θ.Βοσταντζόγλου (1962) ο όρος εξερεύνηση ορίζεται ως "επιτόπια έρευνα αγνώστων μερών". Είναι ιδιαίτερα σημαντικό να τονίσουμε την σημασία της "επιτόπιας έρευνας". Πρέπει δηλαδή να επισκεφθούμε κάποιο συγκεκριμένο σημείο για να βρούμε αυτό που θέλουμε (που στη συγκεκριμένη περίπτωση είναι οι πληροφορίες). Η εξερεύνηση πληροφορίας σε υπολογιστικά συστήματα εξαρτάται κυρίως από δύο παράγοντες:

· **Ποιός έχει την ευθύνη της αναζήτησης της πληροφορίας.**

Στις περιπτώσεις της συνήθους ανάκτησης πληροφοριών, την ευθύνη για την διαδικασία αναζήτησης έχει το σύστημα, το οποίο ασφαλώς γνωρίζει την δομή των αποθηκευμένων πληροφοριών. Αντίθετα, αν την ευθύνη για την αναζήτηση έχει ο χρήστης, πρέπει να είναι ενήμερος και για την δομή των πληροφοριών. Σ' αυτή τη περίπτωση λέμε ότι έχουμε περιήγηση. Για να καταλάβουμε καλύτερα την έννοια της περιήγησης, θα αναφέρουμε την ερώτηση την οποία "θέτει" ουσιαστικά στον χρήστη το σύστημα στην περίπτωση που έχουμε εξερεύνηση πληροφορίας με περιήγηση αλλά και στην αντίθετη περίπτωση. Αν λοιπόν το σύστημα χρησιμοποιεί την περιήγηση στην εξερεύνηση πληροφοριών, η ερώτηση που ο χρήστης καλείται να απαντήσει είναι: "πού θέλεις αμέσως μετά να πας;", ενώ στην αντίθετη περίπτωση η αντίστοιχη ερώτηση είναι: "τι είδους πληροφορίες θέλεις να σου επιστρέψω;". Είναι φανερό ότι στη πρώτη περίπτωση ο χρήστης γνωρίζει τη δομή των πληροφοριών και είναι υπεύθυνος για την αναζήτηση, ενώ στη δεύτερη αρμόδιο για όλα αυτά είναι το σύστημα.

· **Υπαρξη συγκεκριμένου στόχου (πληροφορίας) προς αναζήτηση.**

Το ερώτημα εδώ είναι: "ο χρήστης ψάχνει για κάτι συγκεκριμένο ή ψάχνει στη τύχη;". Στην απάντηση αυτής της ερώτησης βασίζεται η διάκριση μεταξύ των εννοιών *περιόδευση (browsing)* και *υποβολή ερωτήσεων (querying)*. Συγκεκριμένα, αν ο χρήστης έχει κάτι στο μυαλό του, σύμφωνα με το οποίο κάνει την αναζήτηση, τότε μιλάμε για υποβολή ερωτήσεων. Αντίθετα αν η αναζήτηση γίνεται τυχαία χωρίς ο χρήστης να έχει κάτι από πριν στο μυαλό του, μιλάμε για περιόδευση. Όπως προκύπτει από τα παραπάνω, ο ορισμός της περιόδευσης εστιάζεται στη διαφορά μεταξύ της εξερεύνησης πληροφορίας που βασίζεται σε συγκεκριμένο στόχο και της εξερεύνησης πληροφορίας που βασίζεται σε τυχαίες ανακαλύψεις, ενώ ο ορισμός της περιήγησης βασίζεται στην ευθύνη είτε του χρήστη, είτε του συστήματος να γνωρίζουν και να χειρίζονται τη δομή των αποθηκευμένων πληροφοριών. Έτσι, με βάση όσα έχουμε αναφέρει προκύπτει το εξής ερώτημα: Ποιοί από τους παραπάνω τύπους εξερεύνησης πληροφορίας είναι απαραίτητοι σε ένα hypermedia σύστημα;

Κατ' αρχήν, νομίζουμε ότι από τον ορισμό των εννοιών του hypertext και του hypermedia είναι φανερό ότι, τόσο η περιήγηση, όσο και η περιόδευση είναι στοιχεία σύμφυτα με αυτές. Η αποθήκευση των πληροφοριών σε κόμβους και η σύνδεση αυτών με συνδέσμους μάλλον σχεδιάστηκαν με σκοπό έναν καινούριο τρόπο εξερεύνησης πληροφορίας που να συνδυάζει την περιήγηση και την περιόδευση.

Δεν πρέπει όμως να αγνοηθεί και η ανάγκη κάποιων χρηστών για αναζητήσεις συγκεκριμένων και ειδικών τμημάτων πληροφορίας. Όπως καταλαβαίνουμε σε τέτοιες περιπτώσεις η περιήγηση είναι ένας ασύμφορος τρόπος εξερεύνησης πληροφορίας, αφού πέρα από το μεγάλο χρόνο ολοκλήρωσης της αναζήτησης, μπορεί επίσης να υπάρξουν τμήματα

πληροφορίας σχετικά με το αντικείμενο αναζήτησης μας, που να μην τα δούμε. Πρέπει λοιπόν να υπάρχει και η δυνατότητα για υποβολή ερωτήσεων, και μάλιστα με τέτοιο τρόπο ώστε συνεργαζόμενη με τους άλλους τύπους εξερεύνησης πληροφορίας να δίνει την καλύτερη απόδοση για κάθε τύπο αναζήτησης πληροφοριών.

Έχοντας λοιπόν δει, τι εννοούμε με τον όρο εξερεύνηση πληροφορίας και ποιού τύπου εξερεύνησης πληροφορίας ταιριάζουν σε ένα hypermedia σύστημα, θα αναφέρουμε μερικά εργαλεία με τα οποία μπορεί να βοηθηθεί η διαδικασία εξερεύνησης πληροφορίας σε ένα τέτοιο σύστημα.

1.8.5 Browsers

Ένας hypertext browser είναι ένα μέρος του συστήματος (λογισμικό) που δίνει τη δυνατότητα στον χρήστη να δει στην οθόνη του, το διάγραμμα του hypertext δικτύου που ο ίδιος έχει διασχίσει, σε σχέση με ολόκληρο το hypertext δίκτυο.

Μονοπάτια (Trails)

Ένα μονοπάτι είναι μία σειρά κόμβων που ο χρήστης έχει επισκεφθεί κατά τη διάρκεια μίας περιήγησης του στο hypertext δίκτυο του συστήματος.

Περιοδείες (Tours)

Μία περιοδεία είναι ένα μονοπάτι που κάποιος χρήστης μπορεί εύκολα να ακολουθήσει, κάνοντας κάποια επιλογή σε κάποιο **menu**, είτε πατώντας κάποιο **button**.

Bookmarks

Ένα bookmark είναι ένας σημαντικός κόμβος στον οποίο εκχωρούμε κάποιο όνομα ή κάποιο κωδικό έτσι ώστε ανά πάσα στιγμή και από όποιο σημείο και αν βρισκόμαστε μέσω αυτού του κωδικού να μπορούμε να "ταξιδέψουμε" σε αυτόν. Τελειώνοντας αυτή τη παράγραφο θα αναφερθούμε και στο βασικό πρόβλημα του hypertext, τον λεγόμενο **αποπροσανατολισμό (disorientation)** του χρήστη. Αν και ο συσχετιστικός τρόπος με τον οποίο λειτουργεί το hypertext παρέχει μεγάλη ευλυγισία και αυξάνει τη διαθεσιμότητα μεγάλων ποσοτήτων διαφορετικών και ετερογενών πληροφοριών, ταυτόχρονα δημιουργεί και κάποια προβλήματα, αφού η ποικιλία των πληροφοριών μπορεί να οδηγήσει τον χρήστη στο να χάσει κάποιες σημαντικές πληροφορίες που μπορεί να τον ενδιαφέρουν ή ακόμα και να χάσει το προσανατολισμό του μέσα στο σύστημα. Με αυτό εννοούμε ότι ο χρήστης χάνει την αίσθηση του συστήματος, δεν ξέρει σε ποιο σημείο ακριβώς βρίσκεται, ούτε και ποια βήματα ακολούθησε για να φτάσει εκεί. Σε αυτή τη περίπτωση λέμε ότι ο χρήστης είναι **χαμένος ή αποπροσανατολισμένος στο χώρο των πληροφοριών (lost or disoriented in information space)**. Λύση σε αυτό το πρόβλημα μπορούμε να έχουμε με δύο τρόπους, τους οποίους μόλις προηγουμένως αναφέραμε σαν εργαλεία που μπορούν να βοηθήσουν τον χρήστη στη περιήγησή του μέσα σε ένα hypertext δίκτυο. Ο πρώτος από αυτούς είναι η κατασκευή κάποιων χαρτών ή browsers που να επιτρέπουν στο χρήστη να δει ανά πάσα στιγμή σε ποιο τμήμα του hypertext δικτύου βρίσκεται αναφορικά με όλο το δίκτυο καθώς και το μονοπάτι

που ακολούθησε για να βρεθεί εκεί. Ο δεύτερος τρόπος είναι η δυνατότητα κατασκευής κάποιων bookmarks, έτσι ώστε αν ποτέ ο χρήστης χάσει τον προσανατολισμό του να μπορεί να μεταβεί σε κάποιο γνώριμο σε αυτόν σημείο του δικτύου που έχει ορίσει προηγουμένως ως bookmark.

Ο δρόμος προς το hypertext/hypermedia

Πέρα από την ευελιξία και τις τεράστιες δυνατότητες διασύνδεσης πληροφοριών που προσφέρει το hypertext, χαρακτηριστικά τα οποία όπως είπαμε και σε προηγούμενες παραγράφους έπαιξαν σημαντικό ρόλο στην καθιέρωση και εξάπλωση του, κάποιες νέες τεχνολογίες και μια νέα θεώρηση όσον αφορά την αλληλεπίδραση χρήστη-μηχανής που σταδιακά έχουν κάνει την εμφάνιση τους τα τελευταία χρόνια, έχουν επιδράσει αποφασιστικά στην καθολική αποδοχή του, σαν ένα ισχυρό μέσο μοντελοποίησης πληροφορίας. Και βέβαια η καινούρια τάση στην αλληλεπίδραση χρήστη-μηχανής δεν είναι τίποτα άλλο παρά τα γραφικά περιβάλλοντα και τα γραφικά user interfaces, ενώ αναφερόμενοι σε κάποιες καινούριες τεχνολογίες μιλάμε βέβαια για την multimedia τεχνολογία που έχει σημειώσει τεράστια ανάπτυξη τα τελευταία χρόνια και που αποτελεί το κλειδί για το πέρασμα από το hypertext στο hypermedia.

Graphical User Interfaces (GUIs)

Ένα από τα πρώτα ζητήματα στον χώρο της επιστήμης των υπολογιστών ήταν το πως ο χρήστης θα "περάσει" αυτό που θέλει στον υπολογιστή και αντίστοιχα πως θα πάρει την απάντηση του. Μιλάμε δηλαδή για την αλληλεπίδραση χρήστη-συστήματος. Έτσι γεννήθηκε η έννοια του user interface. Οι εντολές ελέγχου του χρήστη και η απόκριση του υπολογιστή σε αυτές αποτελούν το λεγόμενο user interface. Και σκοπός αυτού είναι να διευκολύνει την επικοινωνία χρήστη-υπολογιστή κάνοντας δυσδιάκριτα τα όρια μεταξύ υλικού και λογισμικού. Τα πρώτα user interfaces, αν βέβαια μπορούμε να τα αποκαλέσουμε έτσι περιορίζονταν σε κάποια γραμμή εντολών (**command line**), όπου ο τελικός χρήστης βρισκόταν μπροστά σε ένα εντελώς αφιλόξενο σύστημα και χρειαζόταν να αποστηθίσει ένα μεγάλο αριθμό εντολών προκειμένου να εκτελέσει οποιαδήποτε λειτουργία. Η εποχή αυτή βέβαια έχει περάσει προ πολλού και οι σημερινοί χρήστες έχουν πλέον τεράστιες δυνατότητες χωρίς να χρειάζεται να μπαίνουν σε ειδικές λεπτομέρειες και να αποστηθίζουν ολόκληρα σετ εντολών. Η δυσκολία στη χρήση των υπολογιστών, ο φόβος των χρηστών απέναντι στα δύσχρηστα και αφιλόξενα συστήματα και η χαμηλή παραγωγικότητα που αυτά συνεπάγονταν, ήταν οι παράγοντες που επέβαλλαν την αλλαγή στον τρόπο με τον οποίο ο χρήστης έρχεται σε επαφή με το σύστημα. Και βέβαια η καινούρια τάση στην οποία αναφερόμαστε είναι τα γραφικά user interfaces (Graphical UIs), τα οποία άλλαξαν εντελώς την αντίληψη γύρω από τους υπολογιστές και έφεραν επανάσταση στον τρόπο αλληλεπίδρασης υπολογιστή-χρήστη. Μαζί με τα GUIs αναδείχτηκαν και καθιερώθηκαν κάποια νέα χαρακτηριστικά όπως:

- **Pointing devices** : τα οποία επιτρέπουν στο χρήστη να εστιάσει σε διαφορετικά σημεία της οθόνης είτε για να καλέσει μία εντολή είτε για να χειριστεί κάποια αντικείμενα με διάφορους

τρόπους. Χαρακτηριστικό παράδειγμα βέβαια τέτοιου είδους μηχανισμών είναι το γνωστό σε όλους μας ποντίκι (mouse).

- **Bit-mapped οθόνες** : Έχουν πολύ μεγαλύτερη ανάλυση σε σχέση με τις οθόνες χαρακτήρων, γεγονός που επιτρέπει την επεξεργασία διανυσματικών γραφικών (vector and raster graphics) που επιτρέπουν την παρουσίαση της πληροφορίας στην οθόνη ακριβώς όπως αυτή φαίνεται και στο χαρτί. (πρόκειται για το περίφημο χαρακτηριστικό WYSIWYG - ακρωνύμιο της φράσης What You See Is What You Get - χαρακτηριστικό το οποίο όλοι οι κατασκευαστές user interfaces προσπαθούν να υποστηρίξουν).

- **Παράθυρα (Windows)** : τα οποία επιτρέπουν στους χρήστες την ταυτόχρονη ασχολία με δύο ή περισσότερες εφαρμογές, κάθε μία εκ των οποίων βρίσκεται σε διαφορετικό παράθυρο.

- **Menus** : τα οποία δίνουν τη δυνατότητα στο χρήστη να εκτελέσει κάποιες εντολές πολύ εύκολα χωρίς να χρειάζεται να τις θυμάται.

- **Παράθυρα διαλόγων (Dialog Boxes)** : Δίνουν τη δυνατότητα εισόδου ή εξόδου σύνθετης πληροφορίας και ακόμα τη δυνατότητα εύληπτης επικοινωνίας με το σύστημα.

- **Icons** : Παρέχουν συμβολική αναπαράσταση του συστήματος αλλά και οποιονδήποτε αντικειμένων και τμημάτων αυτού.

Τα GUIs λοιπόν είναι ένας από τους σημαντικότερους λόγους που βοήθησαν στην ριζική αναδιαμόρφωση της εικόνας των υπολογιστών στα μάτια των χρηστών και στην αλλαγή του τρόπου επικοινωνίας χρήστη-μηχανής στα σημερινά συστήματα. Η τάση αυτή για διευκόλυνση του χρήστη και η προσπάθεια που γίνεται, έτσι ώστε ο χρήστης να μπορεί να κάνει όσο περισσότερες εργασίες μπορεί, με τις όσο το δυνατόν λιγότερες ειδικές τεχνικές γνώσεις ήταν όμως από τους λόγους που βοήθησαν και στην ανάδειξη του hypertext. Όπως είπαμε και παραπάνω, το σειριακό κείμενο δεν ανταποκρίνεται στον ανθρώπινο τρόπο σκέψης ενώ το hypertext βρίσκεται πολύ πιο κοντά σ' αυτόν. Βλέπουμε δηλαδή ότι και το hypertext όπως και τα GUIs δημιουργήθηκε μέσα από την σύγχρονη αναγκαιότητα στο χώρο της τεχνολογίας για πιο εύκολα προσπελάσιμα και πιο εύχρηστα ηλεκτρονικά έγγραφα. Τέλος θα πρέπει να αναφέρουμε ότι όλα τα νέα χαρακτηριστικά που τα GUIs καθιέρωσαν είναι από τα βασικά στοιχεία όλων των hypertext/hypermedia συστημάτων.

1.9 Hypermedia : Μέλλον και προοπτικές

Αν και η έννοια του hypertext/hypermedia εμφανίστηκε σχετικά τα τελευταία χρόνια, έχει γνωρίσει πολύ μεγάλη άνθηση και αναγνώριση. Τα συστήματα που ενσωματώνουν hypertext χαρακτηριστικά γίνονται όλο και περισσότερα και ο λόγος γι'αυτό είναι προφανής. Ο συσχετιστικός τρόπος που αυτά λειτουργούν όπως έχουμε αναφέρει και σε προηγούμενες παραγράφους, ταιριάζει πολύ καλύτερα στον ανθρώπινο τρόπο σκέψης με αποτέλεσμα να επιτυγχάνεται πολύ καλύτερη επικοινωνία μεταξύ χρηστών και μηχανής. Επίσης, το γεγονός ότι οι χρήστες μπορούν να προσπελάσουν κάποια πληροφορία με πολλούς τρόπους είναι ένα από τα πιο σημαντικά χαρακτηριστικά των σχετικών συστημάτων. Τα παραπάνω χαρακτηριστικά είναι αυτά που έχουν να καθιερώσουν το hypertext σαν ένα σημαντικό εργαλείο μοντελοποίησης πληροφορίας και αναπαράστασης γνώσης και τα hypermedia συστήματα ως ένα πρότυπο για μία συγκεκριμένη μερίδα των πληροφοριακών συστημάτων

του μέλλοντος. Πρέπει πάντως να τονίσουμε ότι το hypertext δεν είναι μία νέα κατηγορία συστημάτων και εφαρμογών αλλά πρόκειται για μία νέα προσέγγιση που μπορεί να μας βοηθήσει να κάνουμε τις υπάρχουσες εφαρμογές και τα υπάρχοντα συστήματα περισσότερο κατανοητά και ευκολότερα στη χρήση. Βέβαια δεν πρέπει να παραβλέψουμε ότι εκτός από θετικά σημεία υπάρχουν και κάποια προβλήματα, που πρέπει να επιλυθούν έτσι ώστε τα σχετικά συστήματα να έχουν τα περισσότερα από τα επιθυμητά χαρακτηριστικά. Οσον αφορά το μέλλον των hypermedia συστημάτων και την έρευνα που μπορεί να γίνει προς αυτή τη κατεύθυνση, είναι προφανές ότι οι αλλαγές στο χώρο της multimedia τεχνολογίας, επηρεάζουν σημαντικά και το μέλλον των hypermedia συστημάτων. Η ραγδαία ανάπτυξη που παρατηρείται στο χώρο των multimedia υπόσχεται για το μέλλον, ακόμα πιο εντυπωσιακά συστήματα με πολύ περισσότερες δυνατότητες και με ακόμα πιο φιλικό περιβάλλον προς το χρήστη σε τέτοιο βαθμό μάλιστα ώστε το όραμα του Ted Nelson να έχει πάψει προ πολλού να φαντάζει μακρινό.

Κεφ. 2 - Το πρόγραμμα Macromedia Flash

2.1 Εισαγωγή

Το Flash είναι μια σχετικά νέα και επαναστατική πλατφόρμα εφαρμογών διαδικτύου και όχι μόνο, αφού επιτρέπει τη χρήση γραφικών, animation και μουσικής σε μια ιστοσελίδα, χωρίς να επιβαρύνει πολύ το χρόνο φορτώματος της. Με τον τρόπο αυτό οι σχεδιαστές δικτυακών τόπων (web designers) διαθέτοντας εξειδικευμένα εργαλεία είναι σε θέση να δημιουργούν εντυπωσιακές ιστοσελίδες, με κίνηση, εφέ και σύνθετα γραφικά που λειτουργούν με σχετικά καλή ταχύτητα. Εκτός της χρήσης του Flash στο Διαδίκτυο, υπάρχει μια πληθώρα εφαρμογών, από τη δημιουργία μιας παρουσίασης με τη χρήση γραφικών, ήχου και video, μέχρι την ανάπτυξη σύνθετων παιχνιδιών ή διαφόρων προγραμμάτων με ποικίλες εφαρμογές.

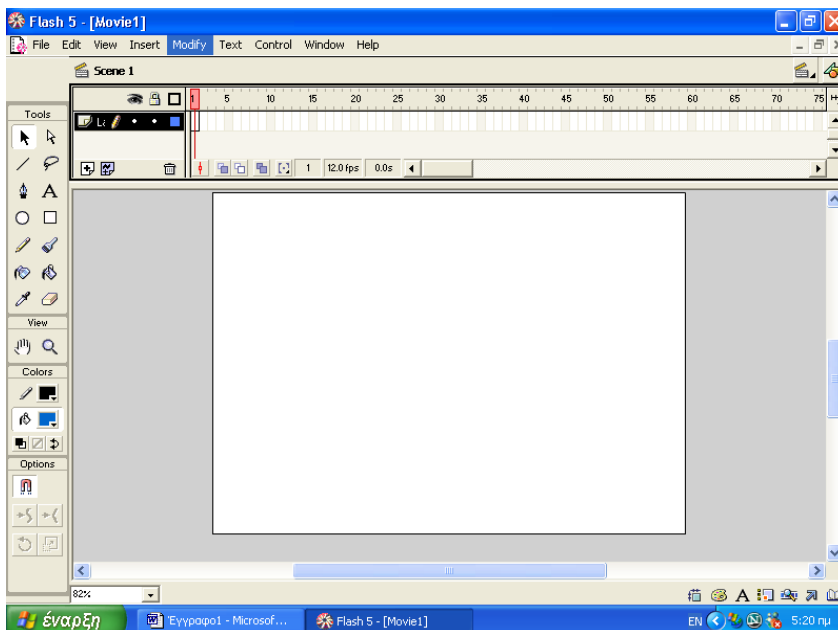
Η γνώση και εφαρμογή των τεχνικών του Flash απαραίτητη προϋπόθεση για τον επαγγελματία πληροφορικής του σήμερα, καθώς εξαιτίας των νέων συνθηκών και της αναγνωρισιμότητας των πολυμεσικών εφαρμογών, η πλειοψηφία των εταιρειών αναζητούν προσωπικό με γνώσεις Flash

2.2 Απλές σημειώσεις για το Flash

Flash και κίνηση

Το φλας είναι δεικαστικό σαν εργαλείο γιατί μας προσφέρει την δυνατότητα δημιουργίας κίνησης γρήγορα κι εύκολα, ενώ ταυτόχρονα μας επιτρέπει να αξιοποιήσουμε την όποια εξοικειώσή μας με τον προγραμματισμό.

Ας αρχίσουμε λοιπόν με κίνηση:



Ανοίξτε ένα καινούργιο αρχείο στο φλας και παρατηρήστε την γραμμή του χρόνου ακριβώς κάτω από την ένδειξη Scene 1 κάτω από τα menu.

Τα κουτάκια της γραμμής λέγονται καρτέ (frames) και αντιπροσωπεύουν τις διαδοχικές «στιγμές» της κίνησης.

Η άδεια σελίδα στο κέντρο της οθόνης είναι η σκηνή μας που μας περιμένει να την γεμίσουμε.

Τέλος η γραμμή των εργαλείων αριστερά θα πρέπει να μας φαίνεται οικεία.

Τα υπόλοιπα ας τα πούμε μέσα από ένα παράδειγμα...

Πριν δημιουργήσετε οτιδήποτε πηγαίνετε στο menu control και διαλέξτε test movie Η άδεια σελίδα που βλέπετε είναι η άδεια σκηνή μας.

Κλείστε την σελίδα δοκιμής για να δείτε ξανά την οθόνη όπου δημιουργείτε.

1. Από τα εργαλεία πάρτε τον κύκλο και δημιουργήστε έναν κύκλο στην πάνω αριστερή γωνία της σκηνής (αν θέλετε επαναλάβετε το test movie).

2. Στην γραμμή χρόνου δείξτε (κάντε κλικ) το κουτάκι κάτω από τον αριθμό τριάντα, και από το menu insert διαλέξτε key frame έτσι θα το μετατρέψτε το από απλό frame , σε frame κλειδί. Πάρτε από τα εργαλεία το βελάκι και σύρτε τον κύκλο κάτω δεξιά. Δείξτε το πρώτο frame και από το μενού insert διαλέξτε create motion tween.

3. Δείξτε το τριακοστό frame και από το μενού insert διαλέξτε create motion tween (ένα βέλος στην γραμμή του χρόνου εμφανίζεται για να δηλώσει την κίνηση). Δοκιμάστε την σκηνή σας.

Καιρός να αναρωτηθείτε και να απαντήσετε πειραματιζόμενοι.

- ✓ Ποιος όρισε την αρχική και την τελική θέση της μπάλας;
- ✓ Ποια εντολή σχεδίασε τις ενδιάμεσες θέσεις;
- ✓ Τι είναι τα key frames ;

Προσπαθήστε τώρα να κάνετε την μπάλα να κινηθεί στην κάτω αριστερή γωνία.

Κίνηση και επίπεδα

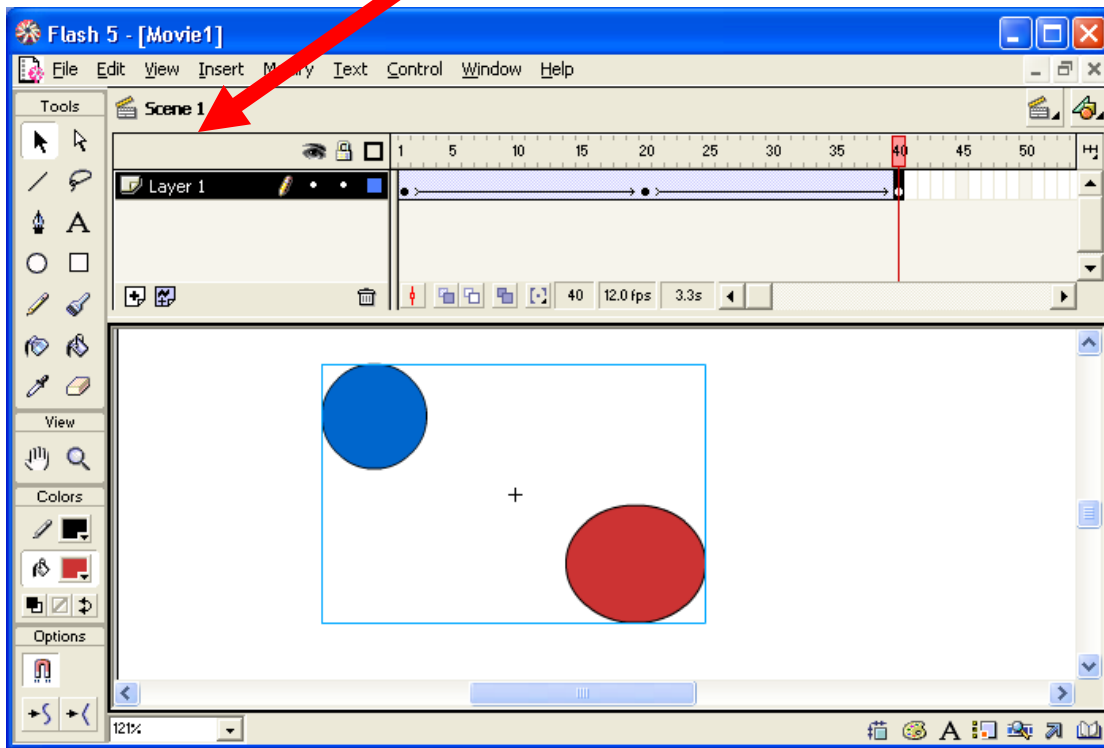
Ζωγραφίστε τώρα δυο μπάλες στην οθόνη και προσπαθήστε να τις κινήσετε.

Μάλλον θα διαπιστώνετε πως οι μπάλες μπορούν να εκτελέσουν μόνο «συντονισμένες» εντελώς όμοιες κινήσεις.

Εκτός κι αν... τις τοποθετήσουμε σε διαφορετικά επίπεδα.

Παρατηρήστε την ταμπέλα «layer1» κάτω από τον χάρακα.

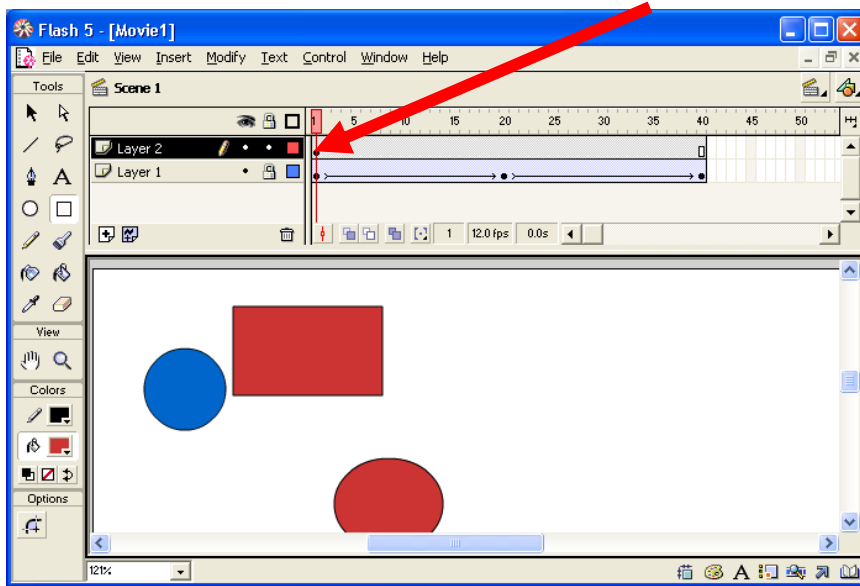
Παρατηρήστε την ταμπέλα «layer1» κάτω από τον χάρακα.



Ας δημιουργήσουμε λοιπόν ένα ακόμη επίπεδο....

Από το μενού insert διαλέξτε layer.

Παρατηρήστε ότι ένα ακόμη επίπεδο με το όνομα «layer2» προστέθηκε πάνω από το «layer1». Κλειδώστε το layer1 και πατήστε στο πρώτο frame του layer2.



Ζωγραφίστε ένα τετράγωνο. Κινήστε το. Θα πρέπει να παρατηρείτε ότι η κίνησή του είναι εντελώς ανεξάρτητη από εκείνη των κύκλων.

Όμως η κίνηση αλλά και τα επίπεδα είναι δυο μεγάλα κεφάλαια στο φλας κι είναι καλύτερα να τα ανακαλύπτουμε ενώ θα μαθαίνουμε άλλα χαρακτηριστικά του εργαλείου.

Σύμβολα και στιγμιότυπα

Σίγουρα ξέρετε τις σφραγίδες που χρησιμοποιούν οι μαθήτρες σας για να γεμίζουν τις σελίδες τους με λουλουδάκια και ζωάκια. Θα'ταν ίσως πιο εύκολο να αγοράζαν ένα ζωγραφιστό λουλούδι, όμως με την σφραγίδα μπορούν να τυπώσουν πολλά ίδια λουλούδια. Όταν πάλι δουλεύατε με την basic σίγουρα θα κάνατε copy paste το ίδιο κουμπί το ίδιο γραφικό ή την ίδια σειρά γραφικών στην ίδια ή σε διαφορετικές φόρμες του ίδιου έργου.

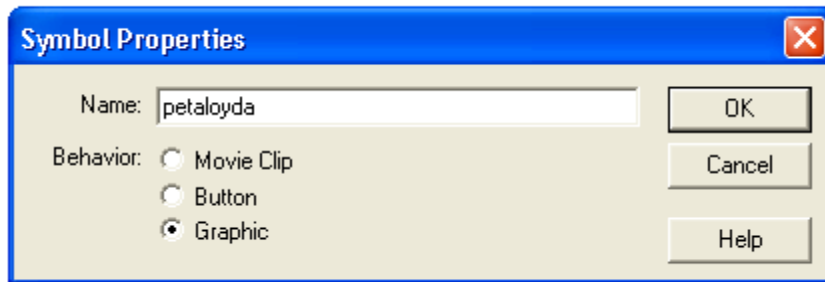
Στο Flash η φιλοσοφία είναι να μην δημιουργείς αντικείμενα μέσα στην σκηνή αλλά σύμβολα σε βιβλιοθήκες κι από κει να σέρνεις όσα στιγμιότυπα θέλεις στις σκηνές σου. (Φανταστείτε τα σύμβολα σαν την σφραγίδα και τα στιγμιότυπα σαν τα λουλουδάκια πάνω στο χαρτί αν και δεν είναι τέλεια η αναλογία).

Έτσι κερδίζεις σε «μέγεθος» (πολύ σημαντικό για δια-δικτυακές εφαρμογές), σε χρόνο αλλά και σε κόπο αφού όσο κι αν φαίνεται παράξενο είναι πιο εύκολο να «ζυγίζεις» και να επεξεργάζεσαι ένα στιγμιότυπο σε μια σκηνή παρά ένα αντικείμενο που δημιούργησες κατευθείαν. Επιπλέον δημιουργείς βιβλιοθήκες γεμάτες αντικείμενα που μπορείς να ξαναχρησιμοποιήσεις σε μελλοντικές εφαρμογές όταν πια θα' χεις ξεχάσει πως τα δημιούργησες, ή να τα δανείσεις σε φίλους και μαθητές.

Στο Flash υπάρχουν τρία είδη συμβόλων: γραφικά κουμπιά και movie clips.

Ένα παράδειγμα σίγουρα θα σας πείσει....

- Ανοίξτε ένα καινούριο αρχείο στο φλας.
- Από το μενού insert διαλέξτε new symbol και στο πλαίσιο που ανοίγει δώστε όνομα petaloyda και επιλέξτε graphic.



- Πατήστε OK. Δίπλα στο όνομα της σκηνής εμφανίζεται το όνομα του γραφικού σας που σας πληροφορεί ότι δεν δουλεύετε πια πάνω στην βασική σκηνή (scene 1) του αρχείου σας, αλλά στο σύμβολο σας.
- Ζωγραφίστε μια πεταλούδα με ανοιχτά φτερά (σ1). Κατόπιν από το edit edit movie (ή πατώντας πάνω στην scene 1 πάνω από την γραμμή του χρόνου) επιστρέψτε στην άδεια!!! Σκηνή σας.
- Από το window library ανοίξτε την βιβλιοθήκη σας και σύρτε την πεταλούδα πάνω στην σκηνή σας. (αν θέλετε δοκιμάστε την σκηνή)

Δημιουργήστε τώρα το graphic petaloyda2, όπου η πεταλούδα σας θα έχει κλειστά φτερά. Σύρτε και μια πεταλούδα με κλειστά φτερά στην σκηνή(σ1).

(εναλλακτικά μπορείτε να κάνετε copy paste την μία από τις δυο έτοιμες πεταλούδες, ή ακόμη καλύτερα να χρησιμοποιήσετε την εντολή file import και να εισάγετε τις πεταλούδες που περιέχονται στα δύο gif αρχεία)

Τώρα θα δημιουργήσετε ένα σύμβολο movie clip

- Από το μενού insert διαλέξτε new symbol και στο πλαίσιο που ανοίγει δώστε όνομα petaloyda_mo και επιλέξτε Movie clip. Πατήστε OK. Δίπλα στο όνομα της σκηνής εμφανίζεται το όνομα του movie clip σας που σας πληροφορεί ότι δεν δουλεύετε πια πάνω στην βασική σκηνή (scene 1) του αρχείου σας, αλλά στο σύμβολο σας.
- Σύρτε στο κέντρο (εκεί που είναι το σταυρουδάκι) το πρώτο γραφικό που δημιουργήσατε (petaloyda).
- Κατόπιν δημιουργήστε ένα key frame στο 10 της γραμμής χρόνου. Σημαδέψτε και σβήστε (del ή cut) την πεταλούδα που βλέπετε (μόνο στο 5^ο frame) και σύρτε την petaloyda2 στο κέντρο. Δημιουργήστε frame στο 10 (insert frame).
- Επιστρέψτε στην βασική σκηνή σας και σύρτε μια petaloyda_mo μέσα.
- Δοκιμάστε την σκηνή.

Μπορείτε να κάνετε την πεταλούδα που ανοιγοκλείνει τα φτερά να πετάει από την μια άκρη της σκηνής στην άλλη: (υπόδειξη:σβήστε τις άλλες πεταλούδες και αφήστε μόνη την πεταλούδα που θέλετε να κινήσετε.)

Κουμπιά

Πριν προχωρήσουμε στα κουμπιά καθαρίστε τα keyframes ως εξής: δείχνετε (κλικ στο τετραγωνάκι) το keyframe και από το insert διαλέγετε clear keyframe. Περιορίστε τα frames σε ένα (σέρνοντας το τετραγωνάκι που εμφανίζεται στο τελευταίο frame)

Καθαρίστε και οτιδήποτε υπάρχει στην σκηνή σας.

Ωρα να δούμε και τα buttons.

- Από το μενού insert διαλέξτε new symbol και στο πλαίσιο που ανοίγει δώστε όνομα petaloyda_bu και επιλέξτε button. Πατήστε OK. Δίπλα στο όνομα της σκηνής εμφανίζεται το όνομα του button σας που σας πληροφορεί ότι δεν δουλεύετε πια πάνω στην βασική σκηνή (scene 1) του αρχείου σας , αλλά στο συμβολό σας
- Προσέξτε ότι, τα τέσσερα πρώτα frame στο κουμπί έχουν όνομα up,over,down, hit.
- Σύρτε στο κέντρο την petaloyda.
- Δημιουργήστε keyframe στο over και σβήστε την petaloyda και φέρτε την petaloyda2.
- Δημιουργήστε keyframe στο down και σβήστε την petaloyda2 και φέρτε την petaloyda_mo.
- *Αν χρειάζεται (???)* Δημιουργήστε keyframe στο hit και ζωγραφίστε ένα τετραγωνάκι που να καλύπτει ολόκληρη την πεταλούδα που υπάρχει από κάτω.
- Επιστρέψτε στην βασική σκηνή , σύρτε μια πεταλούδα κουμπί μέσα **αποθηκεύστε και δοκιμάστε το έργο σας.** (αν θέλετε μπορείτε να σύρτε κι άλλες πεταλούδες από αυτές που υπάρχουν στην βιβλιοθήκη, όμως μην κάνετε κίνηση αυτό θα γίνει αργότερα...)

Βέβαια τα κουμπιά θέλουν κώδικα.....

2.3 Παράδειγμα

Πριν τον προγραμματισμό μια μικρή ανάπαυλα είναι απαραίτητη. Τι θα λέγατε να δείτε και να αξιοποιήσετε **έτοιμα gifs** από αυτά που κατεβάζετε από το δια-δίκτυο;

- Ανοίξτε ένα καινούργιο αρχείο fla αποθηκεύστε το και ονομάστε το: arkoydaki.
- Δημιουργήστε ένα movie clip και ονομάστε το arkoydi_mo(Από το μενού insert διαλέξτε new symbol και στο πλαίσιο που ανοίγει δώστε όνομα arkoydi_mo και επιλέξτε Movie clip. Πατήστε OK. Δίπλα στο όνομα της σκηνής εμφανίζεται το

όνομα του movie clip σας που σας πληροφορεί ότι δεν δουλεύετε πια πάνω στην βασική σκηνή -scene 1- του αρχείου σας , αλλά στο συμβολό σας.)

- Από το μενού file διαλέξτε import και στο πλαίσιο που σας ζητάει όνομα αρχείου εντοπίστε το αρχείο με το αρκουδάκι (ShowLetter το λένε).
- Επιστρέψτε στην βασική σκηνή (edit , edit movie ή κλικ στο όνομα της)
- Ανοίξτε την βιβλιοθήκη (window library) και σύρτε το arkoydi_mo στην σκηνή σας
- Δοκιμάστε την σκηνή σας (control, test movie)
- Επιστρέψτε στο περιβάλλον δημιουργίας.
- Από το μενού file διαλέξτε open as library.
- Στο πλαίσιο που εμφανίζεται διαλέξτε το αρχείο με τις πεταλούδες.
- Από την βιβλιοθήκη που ανοίγει, σύρτε μια πεταλούδα στην σκηνή σας;

Κεφ. 3 – Παιχνίδια που υλοποιήθηκαν

3.1 Ανάλυση και επεξήγηση της εφαρμογής Avoid the cars

Στην εφαρμογή Avoid the cars που υλοποιήσαμε με το Flash το παιχνίδι μας αποτελείται από 4 επίπεδα (**Level 1, Level 2, Level 3, Level 4**) καθώς και 2 καταστάσεις μια όταν ο παίκτης μας κερδίσει και τερματίσει το παιχνίδι (**Game Win**) και μια όταν χάσει το παιχνίδι (**Game Over**).

Τα βασικά συστατικά που εμπεριέχονται και τα βλέπουμε σε όλη την διάρκεια του παιχνιδιού μας είναι ένας μετρητής υγείας (**Health Meter**) ένας μετρητής χρόνου (**Timer**) όπως και ένα **Label** που μας δείχνει κάθε στιγμή σε ποιο επίπεδο παίζουμε. Ξεκινώντας το παιχνίδι ο χειρισμός χαρακτήρα μας του γίνεται με τα βελάκια του keyboard (**Up,Down,Left,Right**) και ο στόχος είναι να περάσει κατά μήκος του αυτοκινητόδρομου με ασφάλεια χωρίς να τον χτυπήσουν τα εισερχόμενα αυτοκίνητα σε άνοδο και κάθοδο.

Στην συνέχεια αφού φτάσει στο τέλος του αυτοκινητόδρομου περνάμε στο επόμενο επίπεδο. Στο **Level 2** υπάρχει παράλληλα και βοήθεια για τον χαρακτήρα μας με **Health Foods** σε τυχαία θέση διάσπαρτα στο αυτοκινητόδρομο και πατώντας πάνω τους ο χαρακτήρας μας ανεβάζει τον δείκτη υγείας του αν είναι τραυματισμένος. Φυσικά περνώντας σταδιακά τα επίπεδα αυξάνονται τα εισερχόμενα αυτοκίνητα και συνολικά έχουμε για το **Level 1** έχουμε 3 cars αριστερά και 1 από δεξιά, για το **Level 2** έχουμε **4 cars** αριστερά και 2 από δεξιά, για το **Level 3** 5 cars αριστερά και 3 από δεξιά και τέλος για το **Level 4** έχουμε 6 cars αριστερά και 4 από δεξιά. Στο τερματισμό του παιχνιδιού έχουμε τις εξής καταστάσεις ο χαρακτήρας μας να χάσει ενδιάμεσα χωρίς να τερματίσει τα 4 επίπεδα όποτε μας εμφανίζει μια κατάσταση **Game Over** και ξανά ξεκινάει το **Level 1** επίπεδο με ένα **Button** που έχει την ένδειξη **Play Again**.

Τέλος αν ο χαρακτήρας μας τερματίσει με επιτυχία το παιχνίδι περνά σε κατάσταση **Game Win** οπότε και βλέπουμε πόσο χρόνο χρειάστηκε για να νικήσει σε seconds.

Στο Παράρτημα Α της εργασίας παρουσιάζεται αναλυτική περιγραφή του κώδικα Action Script της παραπάνω εφαρμογής.

3.2 Ανάλυση και επεξήγηση των επιλογών της εφαρμογής Snakes

Στην εφαρμογή Snakes που υλοποιήσαμε με το Flash το παιχνίδι μας αποτελείται από 3 επίπεδα (**Level 1, Level 2, Level 3**) καθώς και 2 καταστάσεις μια όταν ο παίκτης μας κερδίσει και τερματίσει το παιχνίδι (**Game Win**) και μια όταν χάσει το παιχνίδι (**Game Over**). Τα βασικά συστατικά που εμπεριέχονται και τα βλέπουμε σε όλη την διάρκεια του παιχνιδιού μας είναι ένας μετρητής ζωών (**Lives Left**) ένα **Label** που μας δείχνει στο επίπεδο που παίζουμε πιο **Achievement** όπως και ένα **Label** που μας δείχνει κάθε στιγμή σε ποιο επίπεδο παίζουμε. Ξεκινώντας το παιχνίδι ο χειρισμός χαρακτήρα μας του Snake γίνεται με τα βελάκια του keyboard (**Up,Down,Left,Right**) και ο στόχος είναι να φάει το **Snake 20 Foods** προκειμένου καταρχάς να περάσει στο επόμενο επίπεδο αυξάνοντας την ουρά του με μέγιστο μέγεθος που μπορεί να βρεθεί το **Snake** μας τα 50 Pixel. Περνώντας στο επόμενο επίπεδο το **Snake** μας αυξάνεται τόσο η ταχύτητα του όσο και τα **Foods** που πρέπει να φάει προκειμένου να ολοκληρώσει το επίπεδο και συγκεκριμένα για το **Level 1** πρέπει να φάει 20 **Foods** για το **Level 2** να φάει 40 **Foods** και τέλος για το **Level 3** πρέπει να φάει 60 **Foods** προκειμένου να βρεθεί στην κατάσταση **Game Win**. Για να χάσει το **Snake** μας πρέπει ή να χτυπήσει το Head του στο Wall του παιχνιδιού ή να χτυπήσει το Head του στο Body του **Snake** μας τότε ερχόμαστε στην κατάσταση **Game Over** και το παιχνίδι μας ξανά ξεκινάει πατώντας το Enter το Button Play Again

Στο Παράρτημα Β της εργασίας παρουσιάζεται αναλυτική περιγραφή του κώδικα Action Script της παραπάνω εφαρμογής.

Παράρτημα Α - Παιχνίδι: Avoid the Cars

Παρουσιάζεται ο κώδικας “Action Script” που χρησιμοποιείτε στο συγκεκριμένο παιχνίδι. Υπάρχουν 4 επίπεδα παιχνιδιού (lv1, lv2, lv3, lv4) & Κατάσταση Win & Κατάσταση Gameover.

Επίπεδο «lv1» (Frames 1 – 4)

```
b_speed = 5;

// σταθερή μεταβλητή που χρσιμοποιείται για την εύρεση της ταχύτητας
health = 100;

// κσεκινοντας, η ιγεια είναι στο 100%
_root.startTimer = getTimer();

// Μεταβλητή που वोδάει στον midenismo του xronometrhts kata tin ekkinhsh
//Dimiourgountai ta autokinhta pou 3ekinane apo aristera
function CreateCarLeftToRight(i:Number) {

    //Dimiourgoume ena keno MovieClip pou 8a filoksenhsei to MovieClip tou
autokinhtou

    var        car:MovieClip        =        _root.createEmptyMovieClip("car"+i,
_root.getNextHighestDepth());

    //Episinapetai to MovieClip me identifier car 1 apo th vivlio8hkh
car.attachMovie("car1", "car", car.getNextHighestDepth());

    //Oi sintetagmenes apo tis opoies 8a ksekinhsei to autokinhto.

    //X: -40 kai Y: enas tixaios ari8mos apo ta 40px ws ta 190px (to paxos tou
dromou)

    car._y = Math.round(40+random(150));
    car._x = -40;

    //To tixαιο vima pou 8a akolou8hsei to autokinhto
set("x_dir"+i, random(10)+_root.b_speed);
    car.onEnterFrame = function() {

        //H kinish tou autokinhtou
        car._x = car._x+eval("x_dir"+i);

        //An to autokinhto teleiwsei th diadromh tou, 3ekinai 3ana apo thn arxh,
me kainourgio (tixαιο) vima
```

```

        if (car._x>580) {
            car._y = Math.round(40+random(150));
            car._x = -40;
            set("x_dir"+i, random(10)+_root.b_speed);
        }
        //An to autokinhto xtiphsei ton an8rwpo, afairei 3 apo th zwh tou
        if (car.hitTest(_root.man)) {
            _root.health -= 5;
        }
    };
}
//Dimiourgountai ta autokinhta pou 3ekinane apo deksia. Edw episinaptetai to MovieClip
me identifier car2
//me anti8eto vima apo prin(arnitiko) ka8ws kineitai pros thn anti8eth kateu8insh
function CreateCarRightToLeft(i:Number) {
    var      car:MovieClip      =      _root.createEmptyMovieClip("car"+i,
    _root.getNextHighestDepth());
    car.attachMovie("car2", "car", car.getNextHighestDepth());
    car._y = Math.round(40+random(150));
    car._x = 600;
    set("x_dir"+i, random(10)+_root.b_speed);
    car.onEnterFrame = function() {
        car._x = car._x-eval("x_dir"+i);
        if (car._x<0) {
            car._y = Math.round(40+random(150));
            car._x = 600;
            set("x_dir"+i, random(10)+_root.b_speed);
        }
        if (car.hitTest(_root.man)) {
            _root.health -= 5;
        }
    }
}

```

```

    };
}
//Dhmiourgia twv voh8eiwn health up
function CreateHealth(i:Number) {
    var health:MovieClip = _root.createEmptyMovieClip("health"+i,
_root.getNextHighestDepth());
    health.attachMovie("health", "health", health.getNextHighestDepth());
    health._alpha = 0;
    //To ka8e simvolo topo8eteitai se tixaio shmeio panw sto dromo
    health._y = Math.round(40+random(150));
    health._x = random(520);
    health._width = health._height=17.5;
    health.onEnterFrame = function() {
        //Ksekinontas apo opacity 0, to simvolo health up kanei fade in
        if (health._alpha<=100) {
            health._alpha += 5;
        }
        //Otan o an8rwpos ftasei sto health, an h pros8esh ths igeias tou me to 10
        einai megaitero apo ta 100,
        //h igeia menei sto 100, diaforetika, apla au3anetai kata 10. Epeita, to
        eikonidio eksafanizetai
        if (health.hitTest(_root.man)) {
            if (_root.health+10>100) {
                _root.health = 100;
                removeMovieClip("health"+i);
            } else {
                _root.health += 10;
                removeMovieClip("health"+i);
            }
        }
    };
};

```

```

}
//Afaresh tw n iparxontwn autokinhtwn, svinontas ta MovieClip pou ta filo3enoun
function RemoveTheCars() {
    for (var i = 1; i<=10; i++) {
        removeMovieClip("car"+i);
    }
}
//Afaresh tw n iparxontwn health up, svinontas ta MovieClip pou ta filo3enoun
function RemoveHealth() {
    for (var i = 1; i<=10; i++) {
        removeMovieClip("health"+i);
    }
}
stop();
RemoveTheCars();
// afaresh tw n autokinhtwn
RemoveHealth();
// afaresh tw n health up
//Pros8etoume 4 autokinhta, ek tw n opoion ta 3 phgainoun pros ta deksia kai 1 aristera
for (var i = 1; i<=4; i++) {
    if (i<=3) {
        CreateCarLeftToRight(i);
    } else {
        CreateCarRightToLeft(i);
    }
}
}

```

Επίπεδο «lv2» (Frames 5 – 9)

```
stop();  
RemoveTheCars(); // αφαιresh tw n autokinhtwn  
RemoveHealth(); // αφαιresh tw n health up  
//Pros8etoume 4 autokinhta, ek tw n opoion ta 4 phgainoun pros ta deksia kai 2 aristera  
for (var i = 1; i<=6; i++) {  
    if (i<=4) {  
        CreateCarLeftToRight(i);  
        CreateHealth(i);  
    } else {  
        CreateCarRightToLeft(i);  
    }  
}
```

Επίπεδο «lv3» (Frames 10 – 14)

```
stop();  
RemoveTheCars(); // αφαιresh tw n autokinhtwn  
RemoveHealth(); // αφαιresh tw n health up  
//Pros8etoume 8 autokinhta, ek tw n opoion ta 5 phgainoun pros ta deksia kai 3 aristera  
for (var i = 1; i<=8; i++) {  
    if (i<=5) {  
        CreateCarLeftToRight(i);  
        CreateHealth(i);  
    } else {  
        CreateCarRightToLeft(i);  
    }  
}
```

Επίπεδο «lv4» (Frames 15 – 19)

```
stop();  
RemoveTheCars(); // αφαιresh tw n autokinhtwn  
RemoveHealth(); // αφαιresh tw n health up  
//Pros8etoume 10 autokinhta, ek tw n opoion ta 6 phgainoun pros ta deksia kai 4 aristera
```

```
for (var i = 1; i<=10; i++) {  
    if (i<=6) {  
        CreateCarLeftToRights(i);  
        CreateHealth(i);  
    } else {  
        CreateCarRightToLeft(i);  
    }  
}
```

Επίπεδο «Win» (Frames 20 -25)

stop();

RemoveTheCars(); // afairesh twm autokinhtwn

RemoveHealth(); // afairesh twm health up

Επίπεδο «Game Over» (Frames 25 -31)

RemoveTheCars(); // afairesh twm autokinhtwn

RemoveHealth(); // afairesh twm health up

EXTRA SCRIPT

Υπολογισμός «Total Score»

```
stop();
```

```
totalscore.text = "Your time was "+_root.curTimer+" seconds";
```

Κίνηση Παιδιού

```
// Η kinisi me ta koubakia tou pliktrologiou...
```

```
onClipEvent (enterFrame) {  
    if (Key.isDown(Key.UP)) {  
        this._y = this._y-3;  
    }  
    if (Key.isDown(Key.DOWN)) {  
        this._y = this._y+3;  
    }  
    if (Key.isDown(Key.RIGHT)) {  
        this._x = this._x+3;  
    }  
    if (Key.isDown(Key.LEFT)) {  
        this._x = this._x-3;  
    }  
}
```

Παράρτημα Β - Παιχνίδι: Snake

Παρουσιάζεται ο κώδικας “Action Script” που χρησιμοποιείτε στο συγκεκριμένο παιχνίδι. Υπάρχουν 3 επίπεδα (levels) του παιχνιδιού (level1, level2, level3) & Κατάσταση Win Game & Κατάσταση Gameover.

Frames “1-2-3” Level 1

```
var firstime:Number;

var snakelength:Number;

//Diagrafontai ta hdh iparxonta MovieClip tou body, an iparxoun
for (t=0; t<50; t++) {
    nn = "body"+t;
    removeMovieClip(nn);
}

//Arxikopoioume tous pinakes
posx = new Array();
posy = new Array();

//To arxiko mege8os tou fidiou
snakelength = 10;

//Ksekinontas apo to 210x, gemizoume tous pinakes me tis sintetagmenes tou ka8e
"body"+i
x = 210;
for (t=0; t<snakelength; t++) {
    posx[t] = x;
    posy[t] = 190;
    x += 10;
}

//Auto to xreiazomaste sto 2o frame, gia na topo8ethsoume to fidi stin arxikh tou 8esh
init = 1;

//An h timh ginei 1, sto 2o frame 8a doume ti simvainei. Arxika, h timh einai 0
foods = 0;

//An to swf ekteleitai gia prwth fora, orizoume tis zwes (lives) stis 5, kai to firstime sto
false
```



```

//Etsi ka8e 2h fora pou 8a 3anaer8ei sto frame1, na min 3anakerdisei tis zwes tou
if (firsttime == undefined) {
    lives = 5;
    firsttime = false;
}
//An exei xasei oles tis zwes kai exei dei tin o8onh tou game over, tote oi zwes 3ekinane
3ana apo tis 5
if (lives == 0) {
    lives = 5;
}
//Ta faghta pou exei hdh faei. Arxika den exei faei kanena
totalfoods = 0;
//Ta faghta pou prepei na faei, gia na perasei sto epomeno level
achievement = 20;
stop();



---


//To MovieClip body pollaplasiazetai toses fores oso to arxiko tou mege8os (snakelength
= 10)
if (init == 1) {
    for (t=0; t<snakelength; t++) {
        nn = "body"+t;
        duplicateMovieClip("body", nn, t);
    }
    init = 0;
}
//Pairnontas san arxiko shmeio(simeio anaforas) to kefali tou fidiou (ss. to MovieClip
head)
//kai analoga thn timh ths "up" pou vrisketai mesa sto MovieClip "head", metatopizoume
to MovieClip

```

//kata 10px stin analogh kateu8insh. Etsi, ta MovieClip tou body 8a sinexisoun stin poreia tou kefaliou.

```
if (head.up == 1) {  
    by = head._y;  
    bx = head._x;  
    head._y -= 10;  
}  
if (head.up == -1) {  
    by = head._y;  
    bx = head._x;  
    head._y += 10;  
}  
if (head.left == 1) {  
    by = head._y;  
    bx = head._x;  
    head._x -= 10;  
}  
if (head.left == -1) {  
    by = head._y;  
    bx = head._x;  
    head._x += 10;  
}
```

//An ti prwth timh tou pinaka, einai oi sintetagmenes tou kefaliou, ola ta MovieClp pou apoteloun to swma tou fidiou

//8a sinexisoun stin poreia tou.

```
for (t=snakelength-1; t>0; t--) {  
    posx[t] = posx[t-1];  
    posy[t] = posy[t-1];  
}  
posx[0] = bx;  
posy[0] = by;
```

```

for (t=0; t<snakelength; t++) {
    nn = "body"+t;
    setProperty(nn, _x, posx[t]);
    setProperty(nn, _y, posy[t]);
}
//An to kefali ftasei se enan apo tous 4 toixous, dld oi sintetagmenes tou vgoun e3w apo
ena nohto plaisio pou exoun sximatisei
//ta OR stin parakatw IF, tote an exei dia8esimes zwes, 8a tou afaire8ei mia kai 8a
3ekinhsei 3ana tin pista tin opoia paizei, an oxi,
//oi zwes 8a mhdenistoun k 8a dei tin o8onh game over.
if ((head._x<10) || (head._x>380) || (head._y<50) || (head._y>=460)) {
    if (lives-1>0) {
        lives -= 1;
        _root.gotoAndStop(1);
    } else {
        lives = 0;
        _root.gotoAndStop(10);
    }
}
//Paromoia, an oi sintetagmenes tou head, ginoun ises me kapoia apo tis sintetagmenes
twon MovieClip tou body, simainei pws to fidi
//exei xtiphsei sto swma tou. Ara, omoia me parapanw, n exei dia8esimes zwes, 8a tou
afaire8ei mia kai 8a 3ekinhsei 3ana tin pista tin opoia paizei, an oxi,
//oi zwes 8a mhdenistoun k 8a dei tin o8onh game over.
for (t=2; t<snakelength; t++) {
    if ((head._x == posx[t]) and (head._y == posy[t])) {
        if (lives-1>0) {
            lives -= 1;
            _root.gotoAndStop(1);
        } else {
            lives = 0;

```

```

        _root.gotoAndStop(10);
    }
}
}
//Otan to fidi "vrei kai faei" to faghto tou, diladi an oi sitetagmenes tou head ginoun ises
me tis sitetagmenes tou food, tote h metavlth food (pou mas voh8aei na
epanatopo8ethsoume to
//epomeno) ginetai 0, ta faghta pou exei faei au3anontai kata 1 kai telos, to swma tou
fidiou au3anetai kata 10px, dld kata ena MovieClip body.
if ((food._x == head._x) && (head._y == food._y)) {
    foods = 0;
    totalfoods += 1;
    duplicateMovieClip("body", "body"+snakelength, snakelength);
    snakelength += 1;
    posx[snakelength-2] = posx[snakelength-3];
    posy[snakelength-2] = posy[snakelength-3];
}
//Otan parapanw h timh food ginei 0, simainei pws prepei na epanatopo8ethsoume to
food se diaforetikh tixaia 8esh.
if (foods != 1) {
    x = random(360)+10;
    y = random(320)+50;
    x /= 10;
    y /= 10;
    x = int(x);
    y = int(y);
    x *= 10;
    y *= 10;
    food._x = x;
    food._y = y;
    foods = 1;
}

```

```
}  
//H emfanish twm dia8esimwn zwwn stin o8onh  
LivesLeft = lives;
```

```
//An to totalfood (dld ta faghta pou exei faei) einai iso me to achievement (dld me ta  
faghta pou 8eloume na faei gia na perasei tin pista)  
//ginoun isa, tote 8a dei tin entry scene tis epomenhs pistas, diaforetika, 8a sinexisei tin  
idia, phgainontas 3ana sto frame 2 opou 8a ginei  
//epanato8ethsh tou fidiou stis nees tou 8eseis.  
if (totalfoods<>achievement) {  
    gotoAndPlay(2);  
} else {  
    gotoAndStop(4);  
}
```

Frames “4-5-6” Level 2

```
for (t=0; t<snakelength; t++) {  
    nn = "body"+t;  
    removeMovieClip(nn);  
}  
posx = new Array();  
posy = new Array();  
snakelength = 10;  
x = 210;  
for (t=0; t<snakelength; t++) {  
    posx[t] = x;  
    posy[t] = 190;  
    x += 10;  
}  
init = 1;  
foods = 0;
```

```
totalfoods = 0;
achievement = 40;
stage = 2;
stop();
```

```
if (init == 1) {
    for (t=0; t<snakelength; t++) {
        nn = "body"+t;
        duplicateMovieClip("body", nn, t);
    }
    init = 0;
}
if (head.up == 1) {
    by = head._y;
    bx = head._x;
    head._y -= 10;
}
if (head.up == -1) {
    by = head._y;
    bx = head._x;
    head._y += 10;
}
if (head.left == 1) {
    by = head._y;
    bx = head._x;
    head._x -= 10;
}
if (head.left == -1) {
    by = head._y;
```

```

    bx = head._x;
    head._x += 10;
}
for (t=snakelength-1; t>0; t--) {
    posx[t] = posx[t-1];
    posy[t] = posy[t-1];
}
posx[0] = bx;
posy[0] = by;
for (t=0; t<snakelength; t++) {
    nn = "body"+t;
    setProperty(nn, _x, posx[t]);
    setProperty(nn, _y, posy[t]);
}
if ((head._x<10) || (head._x>380) || (head._y<50) || (head._y>=460)) {
    if (lives-1>0) {
        lives -= 1;
        _root.gotoAndStop(4);
    } else {
        lives = 0;
        _root.gotoAndStop(10);
    }
}
for (t=2; t<snakelength; t++) {
    if ((head._x == posx[t]) and (head._y == posy[t])) {
        if (lives-1>0) {
            lives -= 1;
            _root.gotoAndStop(4);
        } else {
            lives = 0;
            _root.gotoAndStop(10);
        }
    }
}

```

```

        }
    }
}
if ((food._x == head._x) && (head._y == food._y)) {
    foods = 0;
    score = Number(score)+9;
    totalfoods += 1;
    duplicateMovieClip("body", "body"+snakelength, snakelength);
    snakelength += 1;
    posx[snakelength-2] = posx[snakelength-3];
    posy[snakelength-2] = posy[snakelength-3];
}
if (foods != 1) {
    x = random(360)+10;
    y = random(320)+50;
    x /= 10;
    y /= 10;
    x = int(x);
    y = int(y);
    x *= 10;
    y *= 10;
    food._x = x;
    food._y = y;
    foods = 1;
}
LivesLeft = lives;

```

```

if (totalfoods<>achievement) {
    gotoAndPlay(5);
} else {

```



```
        gotoAndStop(7);
    }
```

Frames 7-8-9 “Level 3”

```
for (t=0; t<snakelength; t++) {
    nn = "body"+t;
    removeMovieClip(nn);
}
posx = new Array();
posy = new Array();
snakelength = 10;
x = 210;
for (t=0; t<snakelength; t++) {
    posx[t] = x;
    posy[t] = 190;
    x += 10;
}
init = 1;
foods = 0;
totalfoods = 0;
achievement = 60;
stage = 3;
stop();
```

```
if (init == 1) {
    for (t=0; t<snakelength; t++) {
        nn = "body"+t;
        duplicateMovieClip("body", nn, t);
    }
    init = 0;
}
```

```

if (head.up == 1) {
    by = head._y;
    bx = head._x;
    head._y -= 10;
}
if (head.up == -1) {
    by = head._y;
    bx = head._x;
    head._y += 10;
}
if (head.left == 1) {
    by = head._y;
    bx = head._x;
    head._x -= 10;
}
if (head.left == -1) {
    by = head._y;
    bx = head._x;
    head._x += 10;
}
for (t=snakelength-1; t>0; t--) {
    posx[t] = posx[t-1];
    posy[t] = posy[t-1];
}
posx[0] = bx;
posy[0] = by;
for (t=0; t<snakelength; t++) {
    nn = "body"+t;
    setProperty(nn, _x, posx[t]);
    setProperty(nn, _y, posy[t]);
}

```

```

if ((head._x<10) || (head._x>380) || (head._y<50) || (head._y>=460)) {
    if (lives-1>0) {
        lives -= 1;
        _root.gotoAndStop(7);
    } else {
        lives = 0;
        _root.gotoAndStop(10);
    }
}
for (t=2; t<snakelength; t++) {
    if ((head._x == posx[t]) and (head._y == posy[t])) {
        if (lives-1>0) {
            lives -= 1;
            _root.gotoAndStop(7);
        } else {
            lives = 0;
            _root.gotoAndStop(10);
        }
    }
}
if ((food._x == head._x) && (head._y == food._y)) {
    foods = 0;
    score = Number(score)+9;
    totalfoods += 1;
    duplicateMovieClip("body", "body"+snakelength, snakelength);
    snakelength += 1;
    posx[snakelength-2] = posx[snakelength-3];
    posy[snakelength-2] = posy[snakelength-3];
}
if (foods != 1) {
    x = random(360)+10;

```

```

    y = random(320)+50;
    x /= 10;
    y /= 10;
    x = int(x);
    y = int(y);
    x *= 10;
    y *= 10;
    food._x = x;
    food._y = y;
    foods = 1;
}
LivesLeft = lives;

```

```

//An faei ola ta faghta, tote paei stin win scene, diladi nikhse!

```

```

if (totalfoods <> achievement) {
    gotoAndPlay(8);
} else {
    gotoAndStop(11);
}

```

Frame 10 Game Over

```

stop();
//Diagrafh tou fidiou
for (t=0; t<snakelength; t++) {
    nn = "body"+t;
    removeMovieClip(nn);
}

```

Frame 11 Win Game

```

stop();
//Diagrafh tou fidiou
for (t=0; t<snakelength; t++) {
    nn = "body"+t;
}

```

```
        removeMovieClip(nn);  
    }  
}
```

EXTRA SCRIPT

Achievement (To score)

```
onEnterFrame = function () {  
    //Emfanizei tis plirofories me ta faghta pou exei faei k pou prepei na faei stin  
o8onh  
    DispFoods = _root.totalfoods+" of "+_root.achievement;  
};
```

Head

```
up=0;
left=1;
stop;
```

```
// Edo, exoume ton xeirismo tou fidiou, pou ginetai
// me ta Arrow koubia tou pliktrologiou (up, down, left, right).
//Oi metavlhtes autes diavazontai apo to _root, kai kanoun tis kinhseis opws autes
periegrafhsan
on (keyPress "<Up>") {
    if (up == 0) {
        up = 1;
        if (left == 1) {
            headinhead._rotation = 90;
        } else {
            headinhead._rotation = -90;
        }
        left = 0;
    }
}
on (keyPress "<Down>") {
    if (up == 0) {
        up = -1;
        left = 0;
    }
}
on (keyPress "<Left>") {
    if (left == 0) {
        up = 0;
        left = 1;
    }
}
```

```
    }  
  }  
  on (keyPress "<Right>") {  
    if (left == 0) {  
      up = 0;  
      left = -1;  
    }  
  }  
}
```