

Ηράκλειο, Οκτώβριος 2008

Τ.Ε.Ι ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΠΠ

"ΠΑΡΚΟ ΟΔΙΚΗΣ ΚΥΚΛΟΦΟΡΙΑΣ"

Πτυχιακή Εργασία
Μεμάκης Γεώργιος

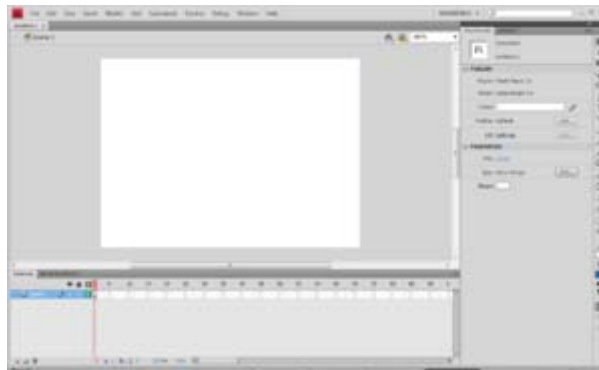
Περιεχόμενα

Περιεχόμενα.....	1
Περιεχόμενα.....	2
1. Εισαγωγή.....	4
1.1 Ιστορία.....	7
1.2 Μελλοντική εξέλιξη.....	10
1.3 Open Screen Project.....	11
1.4 Γλώσσες Προγραμματισμού.....	11
1.5 Ασφάλεια περιεχομένου.....	12
1.6 Action Script.....	13
1.7 Χρονοδιάγραμμα των player.....	14
1.8 Χρονοδιάγραμμα των εκδόσεων της ActionScript.....	16
1.9 Flash Lite.....	18
1.10 Συντακτικό.....	18
1.11 ActionScript 2.0.....	18
1.12 ActionScript 3.0.....	19
1.12.1 Παραδείγματα.....	21
1.13 Δομές Δεδομένων.....	24
1.13.1 Τύποι δεδομένων.....	24
1.13.2 Χρήση των τύπων δεδομένων.....	27
1.14 Ανταγωνισμός.....	30
1.14.1 Format and plug-in.....	30
1.14.2 Το Flash στην αγορά.....	30
1.14.3 Κλασικές εναλλακτικές.....	31
1.15 Τρίτου επιπέδου υλοποίηση.....	32
1.15.1 Λεπτομερής παρουσίαση.....	32

1.15.2 Playback.....	32
1.15.3 Authoring.....	33
1.16 Σχετικά αρχεία τύποι και επεκτάσεις.....	35
1.17 Video στις σελίδες του web	37
1.17.1 Flash Video	37
1.18 Κριτική	39
2. Εφαρμογή.....	40
2.1 Φωτεινοί σηματοδότες	43
2.2 Διαβάσεις πεζών	44
2.3 Πεζοδρόμια	46
2.4 Πινακίδες σήμανσης.....	47
2.5 Διαγραμμίσεις δρόμων	49
2.6 Τροχονόμος	51
3. Ο Κώδικας.....	53
3.1 Κύρια αντικείμενα της σκηνής	53
3.2 Περιοδικές Συναρτήσεις.....	55
3.3 Έλεγχος επιτρεπόμενης κίνησης	56
3.4 Κίνηση Αυτοκινήτου από το πληκτρολόγιο	64
3.5 Αντικείμενο Car	66
3.6 Διαχείριση ήχου	67
3.7 Βιβλιοθήκη Flash	70
4. Επίλογος.....	71
5. Παραρτήματα.....	73

1. Εισαγωγή

Adobe Flash



Adobe Flash CS4 Professional under Windows Vista.

Δημιουργήθηκε από	Adobe Systems (formerly by Macromedia)
Τελευταία release έκδοση	CS4 (10.0.0.544) / 2008-09-23; 20 days ago
Γραμμένο σε	C++
Λειτουργικό Σύστημα	Cross-platform[1]
Τύπος	Multimedia
Άδεια	Proprietary EULA
Website	Adobe Flash Professional Homepage

Το **Adobe Flash** (που ονομαζόταν παλαιότερα **Shockwave Flash** και **Macromedia Flash**) είναι ένα σύνολο λογισμικού πολυμέσων που δημιουργήθηκε από την Macromedia και συχνά αναπτύσσεται και διανέμεται από τα **Adobe Systems**. Από την στιγμή που πρώτο-εμφανίστηκε το **Flash (1996)**, έχει γίνει δημοφιλής μέθοδος για την προσθήκη **animation** και την διαδραστικότητα των ιστοσελίδων. Το **Flash** συνήθιζε να δημιουργεί animation, διαφημίσεις και εξαρτήματα ιστοσελίδων, να ενσωματώνει βίντεο σε ιστοσελίδες και πιο πρόσφατα ανέπτυξε πλούσιες εφαρμογές διαδικτύου.

Το flash μπορεί να χειρίζεται **vector** και **raster graphics** και να υποστηρίζει αμφίδρομη ροή ήχου και βίντεο, και περιέχει μια γλώσσα που ονομάζεται **action script**. Πολλά προϊόντα λογισμικού συστημάτων και συσκευών είναι ικανά να δημιουργούν ή να εμφανίζουν περιεχόμενα του flash, όπως το **Adobe flash player**, το οποίο είναι διαθέσιμο για τους περισσότερους κοινούς web browsers, ορισμένα κινητά τηλέφωνα και άλλες ηλεκτρονικές συσκευές (που χρησιμοποιούν το **Flash Lite**). Το πρόγραμμα δημιουργίας πολυμέσων του **Adobe flash Professional**, χρησιμοποιείτε για να δημιουργήσει το περιεχόμενο για την πλατφόρμα δέσμευσης, όπως οι εφαρμογές, τα παιχνίδια και οι ταινίες και το περιεχόμενο για τα κινητά τηλέφωνα και άλλες ενσωματωμένες συσκευές.

Τα αρχεία με κατάληξη **.swf**, αποκαλούνται παραδοσιακά “shockwave flash” ταινίες “flash movies” ή “flash games” και μπορεί να είναι ένα αντικείμενο μιας ιστοσελίδας, να “παίζεται” σε έναν αυτόνομο flash player, ή ενσωματώνεται σε ένα προβολέα με αυτοτελείς εκτέλεση flash movie (με την .exe επέκταση στα Microsoft Windows). Το flash video (**FLV**) αρχεία έχουν μια **.flv** επέκταση αρχείων, και είτε χρησιμοποιούνται από **.swf** αρχεία, είτε παίζονται μέσω ενός ενήμερου φορέα flv όπως **VLC** είτε το **QuickTime** και το **Windows Media Player** με τα εξωτερικά προστιθέμενα codec’s.

1.1 Ιστορία

Το πρόγραμμα flash ήταν πνευματικό τέκνο του **Jonathan Gay**, ο οποίος ανέπτυξε την ιδέα ενώ ήταν στο κολέγιο, και την επέκτεινε εργαζόμενος για **Silicon Beach Software** και τους διαδόχους του.

Τον Ιανουάριο του 1993 ο Jonathan Gay, ο Charlie Jackson, και ο Michelle Welsh άρχισαν μια μικρή εταιρεία λογισμικού αποκαλούμενη ως **Future Wave** και δημιούργησαν το πρώτο προϊόν το **Smart Sketch**. Μια εφαρμογή των σχεδίων **Smart Sketch** ήταν σχεδιασμένη να δημιουργεί γραφικά τόσο απλά όσο το σχέδιο στο χαρτί. Αρχικά το **Smart Sketch** δεν είχε αποκτήσει σημαντική θέση στην αγορά. Καθώς το διαδίκτυο άρχισε να ανθίζει, η **Future Wave** άρχισε να συνειδητοποιεί τις δυνατότητες για ένα φορέα με βάση το **web-animation** εργαλείο που θα μπορούσε εύκολα να αλλάξει την **Macromedia's Shockwave** τεχνολογία. Το 1995 η **Future Wave** τροποποίησε το **Smart Sketch** με την προσθήκη καρέ-καρέ από **animation** χαρακτηριστικά και ξανά κυκλοφόρησε ως **Future Splash animator** για **PC** και **Macintosh**. Μέχρι τότε η εταιρεία είχε προσθέσει ένα δεύτερο προγραμματιστή τον **Robert Tatsumi** ένα καλλιτέχνη τον **Adam Grofcsik**, και ένα ειδικό τον **Ralph Mittman**. Το προϊόν πρόσφερε πολλά στην **Adobe**, και χρησιμοποιείται από τη **Microsoft** σε πρώιμο έργο με το **Internet (MSN)**. Τον Δεκέμβριο του 1996 η **Macromedia** απέκτησε το φορέα με βάση το **animation software**, και αργότερα κυκλοφόρησε ως **Flash**, contracting "Future" and "Splash" of the FutureWave name.

- **Future Splash animator (10 Απριλίου 1996):** αρχική έκδοση του Flash με βασικά εργαλεία επεξεργασίας και ένα χρονοδιάγραμμα.
- **Macromedia Flash 1 (Νοέμβριος 1996):** μια νέα εμπορική ονομασία Macromedia έκδοση του **Future Splash animator**
- **Macromedia Flash 2 (Ιούνιος 1997):** κυκλοφόρησε με το Flash Player 2, νέα χαρακτηριστικά περιλαμβάνονται: η βιβλιοθήκη αντικειμένων
- **Macromedia Flash 3 (31 Μαΐου 1998):** κυκλοφόρησε με το Flash Player 3, νέα χαρακτηριστικά περιλαμβάνονται: το **movie clip** στοιχείο, **JavaScript plug-in** ενσωμάτωσης, η διαφάνεια και η εξωτερική αυτοδύναμη παίκτη
- **Macromedia Flash 4 (15 Ιουνίου 1999):** κυκλοφόρησε με το Flash Player 4, νέα χαρακτηριστικά περιλαμβάνονται: εσωτερικές μεταβλητές, ένα πεδίο εισαγωγής, **Advanced Action Script**, και **streaming MP3**

- **Macromedia Flash 5 (24 Αυγούστου 2000):** κυκλοφόρησε με το Flash Player 5, νέα χαρακτηριστικά περιλαμβάνονται: **Action Script 1.0** (με βάση **ECMA Script**, γεγονός που καθιστά πολύ παρόμοια με **JavaScript** στην σύνταξη), **XML** υποστήριξη, **Smart clips** (ο πρόδρομος στα συστατικά **Flash**) , **HTML** μορφοποίηση κειμένου προστέθηκαν για δυναμική κείμενο.
- **Macromedia Flash MX (Κατ. 6) (15 Μαρτίου 2002):** κυκλοφόρησε με το **Flash Player 6**, νέα χαρακτηριστικά περιλαμβάνονται: ένα **video codec (Sorenson Spark)**, **Unicode, v1 UI**, συμπίεσης, **Action Script** φορέα κατάρτισης API.
- **Macromedia Flash MX 2004 (version 7) (9 Σεπτεμβρίου 2003):** κυκλοφόρησε με το **Flash Player 7**, νέα χαρακτηριστικά περιλαμβάνονται: **Action script 2.0** (που επέτρεψαν ένα αντικείμενο-μοντέλο για προγραμματισμό **Flash**), συμπεριφορές, επεκτασιμότητας, στρώμα (**JSAPI**), ψευδοκείμενο υποστήριξης, σε χρονολογικά αποτελέσματα.
- **Macromedia Flash MX Professional 2004 (version 7) (9 Σεπτεμβρίου 2003):** κυκλοφόρησε με το **Flash Player 7**, νέα χαρακτηριστικά περιλαμβάνονται στο **Flash MX 2004** (έντυπα για τις μη γραμμικές βασικές καταστάσεις ανάπτυξης και διαφάνειες για την οργάνωση του περιεχομένου σε μια γραμμική μορφή **slide** όπως PowerPoint) δικτυακές υπηρεσίες ολοκλήρωσης, βίντεο εισαγωγής οδηγού, **Media Playback** συνιστώσες (το οποίο ενσωματώνουν ένα πλήρες **MP3** ή / και **FLV player** σε ένα στοιχείο που μπορεί να τοποθετηθεί σε ένα **SWF**), δεδομένα στοιχείων (δεδομένων, **XML Connector**, **Web Services Connector**, **X Update Resolves** , κτλ) και δεσμευτικά δεδομένα , **the Project Panel, v2 UI**, και **Transition class libraries**.
- **Macromedia Flash 8:**
 1. **Macromedia Flash Basic 8 (κυκλοφόρησε στις 13 Σεπτεμβρίου 2005):** Μία λιγότερη πλούσια έκδοση του **Flash** με συγγραφικά εργαλεία που απευθύνονται σε νέους χρήστες που θέλουν μόνο να κάνουν βασική κατάρτιση, ή εμπύχωση ή διαδραστικότητα και κυκλοφόρησε με το **Flash Player 8**, η παρούσα έκδοση του προϊόντος έχει περιορισμένη υποστήριξη για βίντεο και προηγμένα γραφικά και **effects**.

2. **Macromedia Flash Professional 8 (κυκλοφόρησε στις 13 Σεπτεμβρίου 2005):** κυκλοφόρησε με το **Flash Player 8**. Το **Flash Professional 8** επικεντρώθηκε χαρακτηριστικά στην εκφραστικότητα, την ποιότητα βίντεο, στα κινητά και στην συγγραφή. Στις νέες δυνατότητες που περιλαμβάνονται Φίλτρα και αναμιγμένα μέσα, χαλάρωση του ελέγχου για τα κινούμενα σχέδια, **enhanced stroke properties (caps and joins)**, αντικείμενο με βάση το σχέδιο λειτουργίας χρόνου εκτέλεσης bitmap προσωρινής αποθήκευσης, **Flash Type** προηγμένες **anti-aliasing** για το κείμενο, **On2 VP6 advanced video codec**, άλφα υποστήριξη για τη διαφάνεια στο βίντεο, αυθύπαρκτο και προχωρημένο video **encoder** εισαγωγέα, σημείο στήριξης στα **FLV** αρχεία, προηγμένη συνιστώσα αναπαραγωγής βίντεο, και μια διαδραστική κινητή συσκευή **emulator**.
3. **Adobe Flash CS3 Professional (όπως έκδοση 9, κυκλοφόρησε στις 16 Απριλίου 2007):** Το **Flash CS3** είναι η πρώτη έκδοση του **Flash** κυκλοφόρησε υπό την επωνυμία **Adobe**. CS3 χαρακτηριστικά πλήρη υποστήριξη για **Action Script 3.0**, επιτρέπει τα πράγματα να μετατρέπονται σε **Action Script**, προσθέτει καλύτερη ενσωμάτωση με άλλα προϊόντα της **Adobe**, όπως το **Adobe Photoshop**, αι προβλέπει επίσης καλύτερη κατάρτιση **Vector** συμπεριφορά, όλο και περισσότερο όπως **Adobe Illustrator** και το **Adobe Fireworks**.
4. **Adobe Flash CS4 Professional** – είναι η νέα έκδοση του **Flash**, με την ονομασία, "**Diesel**" , δεν πρέπει να συγχέεται με τις τελευταίες δημόσιες prerelease του **Flash Player 10**, με την κωδική ονομασία "**Astro**" που θα περιέχουν αντίστροφη κινηματική (bones), **3D** βασικό αντικείμενο χειραγώγησης, με βάση το αντικείμενο εμπύχωση, ενίσχυση του κειμένου του κινητήρα, και η περαιτέρω επέκταση στην **Action Script 3.0**.

1.2 Μελλοντική εξέλιξη

Τα **Adobe Labs** (πρώην **Macromedia Labs**), είναι πηγή για την πρόωρη προβλέψει σε αναδυόμενες τεχνολογίες και προϊόντα της **Adobe-Macromedia**, συμπεριλαμβανομένης της λήψης του τελευταίου λογισμικού και **plug-ins**. Το **Flash 9**, **Flex 3**, και **Action Script 3,0** συζητούνται στην labs.adobe.com ιστοσελίδα. Μία νέα σημαντική εξέλιξη σε **Flash** (από το 2007) είναι η αυξανόμενη χρήση για την παροχή των επιπέδων παρουσίασης σε συσκευές χειρός. Η Adobe είναι ένα χρυσό βιβλίο για κινητό τηλέφωνο και PDA, και συνεργάζεται για την ανάπτυξη **Flash Lite**, όπως και για το **user interface**.

Το Νοέμβριο του 2007 τα **Adobe Labs** αναπτύσσουν το **Adobe AIR Project** το οποίο είναι ένα **cross-OS runtime** που επιτρέπει στους προγραμματιστές την επαναχρησιμοποίηση των δικτυακών δεξιοτήτων ανάπτυξης (**Flash, Flex, HTML, Ajax**) να χτίζουν και να αναπτύξουν **desktop Rich Internet Applications (Rias)**.

Η επόμενη έκδοση του **Flash** θα έχει δύο επιπλέον συνιστώσες, σχεδιασμένες για μεγάλης κλίμακας εφαρμογές. Αυτό θα ήταν πιο χρήσιμο για μεγάλης κλίμακας **video sites**. Επίσης η **Adobe** έχει ανακοινώσει σχέδια για να προσθέσει **DRM** στη νέα έκδοση του Flash. Με αυτό τον τρόπο η Adobe μπορεί να δώσει στις εταιρείες τη δυνατότητα να συνδέσουν μια διαφήμιση με το περιεχόμενο της και να βεβαιώνουν ότι και οι δύο έχουν παίξει και ότι δεν πρέπει να αλλάξουν.

1.3 Open Screen Project

Την 1η του Μάη, 2008 η **Adobe** ανακοίνωσε την δημιουργία του **Open Screen Project**, που επιθυμεί να οδηγήσει σε πλούσιες εμπειρίες στο Internet και η δημιουργία μιας συνεπούς εφαρμογής διασύνδεσης σε όλες τις συσκευές όπως προσωπικοί υπολογιστές κινητές συσκευές ή ηλεκτρονικά είδη ευρείας κατανάλωσης. Είναι αντιληπτό ότι ο κύριος στόχος του έργου είναι να αναπτύξει το **Adobe Flash** για κινητά τηλέφωνα. Το έργο αυτό έχει πλεονεκτήματα για κατασκευαστές συσκευών, απαλείφει τα τέλη αδειών για **Adobe Flash Player** και ολοκληρωμένο **Runtime Adobe**, καταργεί τους περιορισμούς για τη χρήση του **Shockwave Flash (SWF)** και **Flash Video (FLV)** μορφές αρχείων και δημοσιεύει **Application Programming Interfaces** για την φορητότητα Flash σε νέες συσκευές. Δημοσιεύει, επίσης, **Flash Cast protocol** και **Action Message Format (AMF)**, τα οποία αφήνουν τις εφαρμογές flash να λαμβάνουν πληροφορίες από απομακρυσμένες βάσεις δεδομένων.

1.4 Γλώσσες Προγραμματισμού

Αρχικά επικεντρώθηκε σε κινούμενα σχέδια Flash, και πρώτες εκδοχές του περιεχομένου που προσφέρει μερικά χαρακτηριστικά της διαδραστικότητας και συνεπώς, είχε πολύ περιορισμένη δυνατότητα scripting.

Πιο πρόσφατες εκδόσεις περιλαμβάνουν **Action Script**, η εφαρμογή του **ECMA Script** πρότυπο το οποίο, ως εκ τούτου, έχει την ίδια σύνταξη όπως **JavaScript**, αλλά σε διαφορετικό πλαίσιο προγραμματισμού που συνδέονται με ένα διαφορετικό σύνολο της κατηγορίας των βιβλιοθηκών. Η **Action Script** χρησιμοποιείται για να δημιουργήσει σχεδόν το σύνολο της διαδραστικότητας (κουμπιά, πεδία κειμένου εισόδου, drop down μενού) που παρατηρείται σε πολλές εφαρμογές Flash.

Το **Flash MX 2004** εισήγαγε την **Action Script 2.0**, μία **scripting** γλώσσα προγραμματισμού που να ταιριάζει περισσότερο στην ανάπτυξη των εφαρμογών Flash. Συχνά είναι δυνατή η εξοικονόμηση χρόνου από **scripting** και όχι στην εμπύχωση του, που συνήθως επιτρέπει επίσης υψηλότερο επίπεδο σχεδιασμού. Με την άφιξη του **Flash Player 9 alpha** μια νεότερη έκδοση του έχει κυκλοφορήσει **Action Script, Action Script 3.0**. Η **Action Script 3.0** είναι ένα αντικείμενο προσανατολισμένης γλώσσας προγραμματισμού που επιτρέπει για περισσότερο έλεγχο και δυνατότητα επαναχρησιμοποίησης κώδικα για την δημιουργία σύνθετων Flash εφαρμογών.

Η **Action Script 3.0** έχει επίσης τη δυνατότητα για επίσημη Τεχνολογία Λογισμικού μεθόδους που θα εφαρμοστή κατά την εργασία με το **Flash**, λόγω του αντικειμένου προσανατολισμένης προσέγγισης του προγραμματισμού.

Τον τελευταίο καιρό, οι **Flash libraries** είναι να χρησιμοποιηθούν με **XML** δυνατότητες του **browser**, να καταστούν πλούσια σε περιεχόμενο στο πρόγραμμα περιήγησης. Η τεχνολογία αυτή είναι γνωστή ως ασύγχρονη **Flash** και **XML**, όπως **AJAX**. Αυτή η τεχνολογία της **Asynchronous Flash** και **XML** άσκησε πίεση για μια πιο επίσημη προσέγγιση αυτής της τεχνολογίας που ονομάζεται **Adobe Flex**, που χρησιμοποιεί το **Flash runtime** να οικοδομήσει τις **Rich Internet Applications**.

Αυτή η τεχνολογία μπορεί να χρησιμοποιηθεί σε παίκτες όπως αυτόν της **MySpace** και το **YouTube**, για την παροχή προστασίας για το περιεχόμενο που ζητεί το Flash, όπως MP3s και video. Το περιεχόμενο είναι συνεχής ροή ή περνάει μέσω των αρχείων Flash, κάνοντας το κατέβασμα ένα δύσκολο έργο για τους περισσότερους ανθρώπους. Προγράμματα όπως το **Real Player Downloader browser** και επεκτάσεις όπως **Firebug** μπορεί να εντοπίζουν τα αρχεία **XML**.

1.5 Ασφάλεια περιεχομένου

Συχνά οι δημιουργοί Flash εφαρμογών παρόλο που επιθυμούν τα πλεονεκτήματα που τους παρέχει το Flash στον τομέα του animation και της διαδραστικότητας, δεν επιθυμούν να εκθέσουν τις εικόνες ή / και κώδικα για τον κόσμο. Ωστόσο, μέσα από το **.SWF** αρχείο που είναι αποθηκευμένο σε τοπικό επίπεδο, μπορεί κανείς να επιχειρήσει στη συνέχεια να κάνει **decompile** και να αναπαράγει τον πηγαίο κώδικα καθώς και άλλα στοιχεία (όπως αντικείμενα της βιβλιοθήκης). Κάποιοι **decompilers** είναι ικανοί να ανακατασκευάσουν σχεδόν πλήρως το αρχικό αρχείο, να προβλέπεται ο κώδικας που χρησιμοποιήθηκε κατά τη διάρκεια της δημιουργίας, ακόμα και αν τα αποτελέσματα ποικίλουν σε κάποια περίπτωση.

Κατά τους **decompilers**, δημιουργήθηκαν εφαρμογές των **.SWF** που παρέχουν μερική ασφάλεια. Η υψηλή ποιότητα χρησιμοποιεί παγίδες για τους **decompilers**, δημιουργώντας λάθη, αλλά κανένας δεν έχει οριστικά αποδειχθεί ικανός για την προστασία όλων των περιεχομένων.

1.6 Action Script

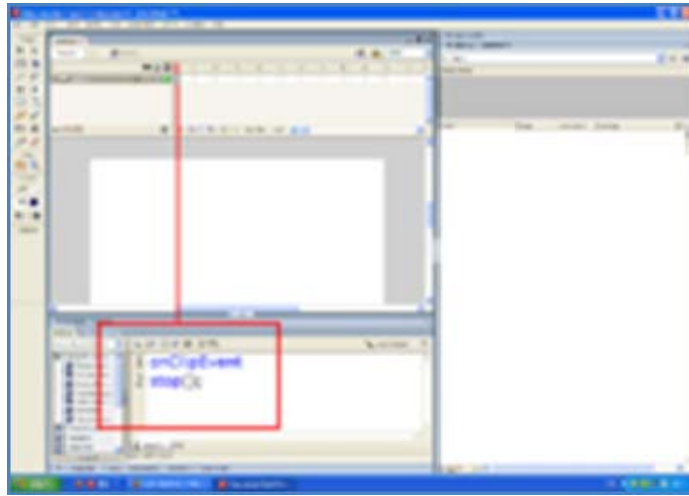
Η **Action Script** είναι μια γλώσσα με βάση **ECMAScript**. Η **Action Script** χρησιμοποιείται κυρίως για την ανάπτυξη των websites και τη χρήση του λογισμικού **Adobe Flash Player platform** (με τη μορφή SWF αρχεία ενσωματωμένα σε ιστοσελίδες), αλλά είναι επίσης βάση δεδομένων που χρησιμοποιείται σε ορισμένες εφαρμογές (όπως η Alpha Five). Αρχικά αναπτύχθηκε από την Macromedia, η γλώσσα είναι πλέον ιδιοκτησία της Adobe (Macromedia που αποκτήθηκαν το 2005). Η **Action Script** είχε αρχικά σχεδιασθεί για τον έλεγχο του φορέα απλό **2D animation** σε Adobe Flash (πρώην Macromedia Flash). Οι πρόσθετες νεότερες εκδόσεις λειτουργίες που επιτρέπουν τη δημιουργία παιχνιδιών που βασίζονται στο Web και πλούσιες Internet εφαρμογές με streaming media (όπως το βίντεο και ήχου).

Η **Action Script** ξεκίνησε ως **scripting** γλώσσα για την **Macromedia Shockwave Flash** συγγραφικά εργαλεία, που αναπτύχθηκε από την Adobe ως Adobe Flash. Οι τρεις πρώτες εκδόσεις του Flash παρέχουν συγγραφικά εργαλεία με περιορισμένες δυνατότητες διαδραστικότητας. Οι προγραμματιστές θα μπορούσαν να αποδίδουν μια απλή εντολή, που ονομάζεται "**action**", σε ένα κουμπί ή ένα **frame**. Το σύνολο των ενεργειών ήταν η βασική έλεγχοι πλοήγησης, με εντολές όπως "**play**", "**stop**", "**getURL**", και "**gotoAndPlay**".

Με την κυκλοφορία του **Flash 4** το 1999, αυτό το απλό σύνολο των δράσεων έγινε μια μικρή scripting γλώσσα. Οι νέες δυνατότητες που εισήχθησαν για Flash 4 περιλαμβάνουν μεταβλητές, εκφράσεις, φορείς, δηλώσεις, και βρόχους. Αν και αναφέρεται ως "**Action Script**", το Flash 4 εγχειρίδιο χρήσης και τα εμπορικά έγγραφα συνέχισαν να χρησιμοποιούν τον όρο "ενέργειες" για να περιγράψει αυτό το σύνολο των εντολών.

1.7 Χρονοδιάγραμμα των player

- **Flash Player 2:** Η πρώτη έκδοση με υποστήριξη για δέσμες ενεργειών. Οι δράσεις που περιλαμβάνονται **gotoAndPlay, gotoAndStop, nextFrame και nextScene** χρονοδιάγραμμα για έλεγχο.
- **Flash Player 3:** Η εκτεταμένη βασική υποστήριξη για δέσμες ενεργειών με την ικανότητα της φόρτωσης εξωτερικού **SWFs (load Movie)**.
- **Flash Player 4:** Πρώτος παίκτης με μια πλήρη εφαρμογή scripting (που ονομάζεται Ενέργεια). Οι δέσμες ενεργειών ήταν ένα **slash based syntax** και η υποστήριξη περιέχεται για βρόχους, conditionals, μεταβλητές και άλλες βασικές γλωσσικές δομές.
- **Flash Player 5:** Περιλαμβάνεται η πρώτη έκδοση του **Action Script**. Χρησιμοποιείται ένα πρότυπο με βάση το προγραμματισμό με βάση **ECMAScript**, και επιτρέπονται όλες οι διαδικασίες προγραμματισμού και αντικείμενα προγραμματισμού με προσανατολισμό
- **Flash Player 6:** Προστέθηκε ένα γεγονός μοντέλου χειρισμού, ελέγχου και υποστήριξης της προσβασιμότητας για διακόπτη. Η πρώτη έκδοση με υποστήριξη για την AMF RTMP και πρωτόκολλα, τα οποία επιτρέπεται για **on demand audio / video streaming**.
- **Flash Player 7:** Οι προσθήκες περιλαμβάνουν **CSS styling** για το κείμενο και τη στήριξη των **Action Script 2.0**, μία γλώσσα προγραμματισμού με βάση την **ECMAScript 4 Netscape** πρόταση για βασική κλάση κληρονομιάς. Ωστόσο, η **Action Script 2,0** μπορούν να καταρτίσουν **Action Script 1,0 byte-code**, ώστε να μπορεί να τρέχει σε Flash Player 6.
- **Flash Player 8:** έχει επεκταθεί περαιτέρω **Action Script 1/Action Script 2** με την προσθήκη νέας κατηγορίας βιβλιοθήκης με API για τον έλεγχο των δεδομένων σε **bitmap** χρόνου εκτέλεσης, upload αρχείων και ζωντανά φίλτρα για την θαμπάδα και **drop shadow**



Παράδειγμα Action Script 2.0, τρέχει στο Macromedia Flash 8.

- **Flash Player 9** (αρχικά ονομαζόταν 8.5): Η προσθήκη **Action Script 3.0** με την έλευση μιας νέας εικονικής μηχανής, που ονομάζεται **AVM2 (Action Script Virtual Machine 2)**, που συνυπάρχει με το προηγούμενο **AVM1** που απαιτούνται για να υποστηρίξουν την κληρονομιά περιεχόμενο. Η αύξηση απόδοσης ήταν ένας βασικός στόχος για αυτή την αποδέσμευση του παίκτη που περιλαμβάνει ένα νέο **JIT compiler**. Αυτή είναι η πρώτη αποδέσμευση του παίκτη για να πάρει τον τίτλο Adobe Flash Player.
- **Flash Player 10 beta** (αρχικά ονομαζόταν Astro): Προστέθηκαν βασικές 3D χειρισμοί, όπως η περιστροφή στον X, Y, Z άξονα, και ένα σχέδιο 3D API. Ικανότητα για δημιουργία custom φίλτρων χρησιμοποιώντας το **Adobe Pixel Bender**. Η γραφική επεξεργασία είναι τώρα που εξέρχεται με την GPU που δίνει μια αισθητή αύξηση του χρόνου να καταστεί για κάθε frame, ουσιαστικά υψηλότερη ταχύτητα frame. Υπάρχει ένας νέος ήχος API που επιτρέπει τη προσαρμοσμένη δημιουργία ήχου σε φλας, κάτι που ποτέ δεν ήταν δυνατή πριν.

1.8 Χρονοδιάγραμμα των εκδόσεων της ActionScript

2000-2003: Η **Action Script "1.0"** με την κυκλοφορία του Flash 5 το Σεπτέμβριο του 2000, τα "actions" από Flash 4 ενισχύθηκαν για μια ακόμη φορά και το όνομα "**Action Script**" για πρώτη φορά [3]. Αυτή ήταν η πρώτη έκδοση του Action Script με επιρροές από JavaScript και η ECMA-262 (Τρίτη Έκδοση) πρότυπο, που υποστηρίζουν το πρότυπο του μοντέλου αντικειμένου και πολλοί από τους βασικούς τύπους δεδομένων. Οι τοπικές μεταβλητές μπορούν να δηλώνονται με την δήλωση **var**. Συγκεκριμένα, η Action Script θα μπορούσε πλέον να πληκτρολογείτε με ένα πρόγραμμα επεξεργασίας κειμένου και όχι να συναρμολογούνται από την επιλογή δράσεων από drop-down λίστες διαλόγου και ελέγχου. Με την επόμενη έκδοση του εργαλείου συγγραφής, Flash MX, και τον αντίστοιχο Flash Player 6, η γλώσσα παρέμεινε ουσιαστικά αμετάβλητη υπήρχαν μόνο μικρές αλλαγές, επιπλέον, όπως η δήλωση του διακόπτη και την "**strict equality**" φορέα, που την έφερε πιο κοντά στο να ECMA-262 συμμορφώνεται. Δύο σημαντικά χαρακτηριστικά της **Action Script** που την κάνει να ξεχωρίζει από τις νεότερες εκδόσεις είναι ο χύμα τύπος συστήματος και η στήριξή τους σε πρωτότυπα με βάση το κληρονομιά. Η άνετη πληκτρολόγηση αναφέρεται στην ικανότητα μιας μεταβλητής να κατέχει οποιοδήποτε είδος των δεδομένων. Αυτό επιτρέπει τη γρήγορη ανάπτυξη σεναρίου και είναι κατάλληλο για μικρής κλίμακας έργα scripting. Η πρωτότυπη βάση κληρονομιάς είναι η **Action Script 1.0** μηχανισμός για την επαναχρησιμοποίηση κώδικα και αντικείμενο προγραμματισμού με προσανατολισμό. Αντί της κλάσης **key-word** που καθορίζει κοινά χαρακτηριστικά της κλάσης, η Action Script 1,0 χρησιμοποιεί ένα ειδικό αντικείμενο που λειτουργεί ως "πρότυπο" για μια κατηγορία αντικειμένων. Όλα τα κοινά χαρακτηριστικά μιας κλάσης ορίζονται στην κλάση του πρωτοτύπου αντικείμενο και κάθε στιγμή αυτής της κλάσης περιέχει μια σύνδεση με το πρωτότυπο αντικείμενο.

2003-2006: **Action Script 2.0** η επόμενη αναθεώρηση της γλώσσας, η **Action Script 2.0**, εισήχθη το Σεπτέμβριο του 2003 με την κυκλοφορία του **Flash MX 2004** και το αντίστοιχο player, το **Flash Player 7**. Ανταποκρινόμενη στη ζήτηση των χρηστών για μια γλώσσα καλύτερα εξοπλισμένη για τις μεγαλύτερες και πιο πολύπλοκες εφαρμογές, που εμφανίζουν Action Script 2.0 χρόνο, συγκεντρώνει το είδος ελέγχου και τάξης που βασίζεται στην σύνταξη, όπως οι λέξεις-κλειδιά της κλάσης και επεκτάσεις. (Αν και αυτό επιτρέπεται για ένα πιο ευέλικτο αντικείμενο προσανατολισμένης προσέγγισης του προγραμματισμού, ο κώδικας θα εξακολουθεί να καταρτίζεται σε Action Script 1.0 byte code, ώστε να μπορούν να χρησιμοποιηθούν για την προηγούμενη έκδοση Flash Player 6, τόσο καλά. Με άλλα

λόγια, η τάξη με βάση την κληρονομική σύνταξη ήταν ένα στρώμα επάνω από το υπάρχον σύστημα που βασίζεται στο πρωτότυπο.) Με την Action Script 2.0, οι προγραμματιστές μπορούσαν να περιορίσουν τις μεταβλητές με ένα συγκεκριμένο τύπο με την προσθήκη ενός είδους σχολιασμό ώστε ο τύπος αντιστοιχίας σφαλμάτων θα μπορούσε να βρεθεί σε compile-time. Η **Action Script 2.0** εισήγαγε επίσης την κατηγορία με βάση την κληρονομική σύνταξη, έτσι ώστε οι προγραμματιστές θα μπορούσαν να δημιουργήσουν τάξεις και interfaces, όσο και στην κλάση που θα βασίζεται σε γλώσσες όπως **Java** και **C + +**. Αυτή η έκδοση εν μέρει προσαρμόστηκε με την τέταρτη έκδοση **ECMAScript** σχέδιο προδιαγραφών.

2006-σήμερα: Η **Action Script 3.0** τον Ιούνιο του 2006 έκανε ντεμπούτο με το **Adobe Flex 2.0** και το αντίστοιχο player, το Flash Player 9. Η **Action Script 3.0** ήταν μια θεμελιώδη αναδιάρθρωση της γλώσσας, μέχρι του σημείου να χρησιμοποιεί μια εντελώς διαφορετική εικονική μηχανή. Ο **Flash Player 9** περιέχει δύο εικονικές μηχανές, για AVM1 γραμμένο σε κώδικα **Action Script 1.0** και **2.0**, και για το περιεχόμενο AVM2 γραμμένο σε **Action Script 3.0**. Η **Action Script 3.0** είναι ταχύτερη από τις προηγούμενες εκδόσεις αλλά δεν είναι ακόμη αρκετά γρήγορη για εφαρμογές **full screen** με πολλά αντικείμενα που δραστηριοποιούνται. Η **Action script 3.0** έχει περιορισμένη υποστήριξη για την επιτάχυνση υλικού (DirectX, OpenGL).

Η ενημέρωση για τη γλώσσα, εισήγαγε αρκετά νέα χαρακτηριστικά:

- **Compile-time** και **runtime** τύπος του τύπου ελέγχου υπάρχουν πληροφορίες και για τα δυο, Compile-time και runtime.
- Βελτιωμένη απόδοση από μια κλάση, με βάση το κληρονομικό σύστημα ξεχωριστό από το μοντέλο με βάση το κληρονομικό σύστημα.
- Υποστήριξη για πακέτα, περιοχές, καθώς και τακτικές εκφράσεις.
- Ομαδοποιεί σε ένα εντελώς νέο είδος byte code, ασυμβίβαστη με Action Script 1.0 και 2.0 byte code.
- Αναθεωρημένη Flash Player API, οργανώνετε σε πακέτα.
- Ενοποιημένη περίπτωση χειρισμού του συστήματος που βασίζεται στο πρότυπο DOM περίπτωση χειρισμού.
- Ένταξη των ECMAScript για XML (E4X), για τους σκοπούς της επεξεργασίας XML.

- Άμεση πρόσβαση στο Flash runtime screen director για πλήρη έλεγχο του τι γίνεται που εμφανίζεται κατά την εκτέλεση.
- Η πλήρης εφαρμογή του σύμφωνου ECMAScript τέταρτη έκδοση Σχέδιο προδιαγραφών.

1.9 Flash Lite

- **Flash Lite 1.0:** Flash Lite είναι η τεχνολογία Flash αναπτύχθηκε ειδικά για κινητά τηλέφωνα και καταναλωτικές ηλεκτρονικές συσκευές. Το Flash 4 υποστηρίζει Action Script.
- **Flash Lite 1.1:** Flash 4 Action Script υποστηρίζει, και προσθέτη συσκευή APIs προσθήκης .
- **Flash Lite 2.0 και 2.1:** Προστιθέμενη υποστήριξη για Flash 7 ActionScript 2.0 και κάποιες πρόσθετες fsccommand2 API.
- **Flash Lite 3:** Πρόσθετη υποστήριξη για Flash 8 ActionScript 2.0 και επίσης FLV αναπαραγωγής βίντεο.

1.10 Συντακτικό

Η **Action Script code** είναι ελεύθερη μορφή και έτσι μπορεί να δημιουργηθεί με όποιο ποσό ή στυλ του **white space**, ότι ο συγγραφέας επιθυμεί. Η βασική σύνταξη προέρχεται από ECMAScript.

1.11 ActionScript 2.0

Ο παρακάτω κώδικας ,ο οποίος δουλεύει σε οποιαδήποτε συμβατό player, δημιουργεί ένα πεδίο κειμένου σε βάθος 0, στη θέση (0, 0) επί της οθόνης (που μετριέται σε rixel), που είναι 100 rixels πλάτος και ύψος .Στη συνέχεια η παράμετρος του κειμένου έχει ρυθμιστεί με το "Hello, world!" String, και είναι αυτόματα στον παίκτη:

```
createTextField("greet", 0, 0, 0, 100, 100);  
greet.text = "Hello, world";
```

When writing external ActionScript 2.0 class files the above example could be written in a file named Greeter.as as following.

```
class com.example.Greeter extends MovieClip{  
    public function Greeter() {}  
    public function onLoad() :Void
```

```
{
    var txtHello:TextField = this.createTextField("txtHello", 0,
0, 0, 100, 100);
    txtHello.text = "Hello, world";
}
}
```

1.12 ActionScript 3.0

Η **Action Script 3.0** έχει παρόμοια σύνταξη, με την **Action Script 2.0** αλλά ένα διαφορετικό σύνολο APIs για τη δημιουργία αντικειμένων. Συγκρίνετε τα παρακάτω με την προηγούμενη έκδοση **Action Script 2.0**:

```
var greet:TextField = new TextField();
greet.text = "Hello World";
this.addChild(greet);
```

Minimal Action Script 3.0 τα προγράμματα μπορεί να είναι κάπως μεγαλύτερα και πιο περίπλοκα λόγω της αύξησης του διαχωρισμού της γλώσσας προγραμματισμού και το Flash IDE.

Τεκμηριώνουν το ακόλουθο αρχείο για να Greeter.as:

```
package com.example
{
    import flash.text.TextField;
    import flash.display.Sprite;

    public class Greeter extends Sprite
    {
        public function Greeter()
        {
            var txtHello:TextField = new TextField();
            txtHello.text = "Hello World";
            addChild(txtHello);
        }
    }
}
```

```
    }  
  }  
}
```

Τέλος, ένα παράδειγμα χρήσης Action Script κατά την ανάπτυξη Flex εφαρμογών, και πάλι τεκμηριώνει το ακόλουθο περιεχόμενο να είναι σε αρχείο με το όνομα Greeter.as:

```
package  
{  
    public class Greeter  
    {  
        public static function sayHello():String  
        {  
            var greet:String = "Hello, world!";  
            return greet;  
        }  
    }  
}
```

Ο κωδικός αυτός θα συνεργαστεί με τις ακόλουθες MXML φάκελο αίτησης:

```
<?xml version="1.0" encoding="utf-8"?>  
  
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" xmlns="*"   
layout="vertical" creationComplete="initApp()">  
  
    <mx:Script>  
        <![CDATA[  
            public function initApp():void  
            {
```

```
        // Prints our "Hello, world!" message
into "mainTxt".

        mainTxt.text = Greeter.sayHello();
    }

    ]]>

</mx:Script>

<mx:Label id="title" fontSize="24" fontStyle="bold"
text="Hello, world!" Example' />

<mx:TextArea id="mainTxt" width="250" />

</mx:Application>
```

1.12.1 Παραδείγματα

ActionScript 2.0 παραδείγματα

Οι παρακάτω κώδικα τυπώνει Hello world στο stage (screen).

```
this.createTextField("myText", 1, 0, 0, 100, 25);
this.myText.text = "Hello world!";
```

Ο παρακάτω κώδικας στην έξοδο στην τρέχουσα θέση του ποντικιού όταν το ποντίκι κινείται, χρησιμοποιώντας το **onMouseMove event**. Και πάλι αυτό μπορεί να λειτουργήσει μόνο στο Flash IDE.

```
onMouseMove = function ()
{
    trace("X: " + this._xmouse);
    trace("Y: " + this._ymouse);
};
```

Ο παρακάτω κώδικας, όταν τοποθετείται ένα κουμπί, αρχίζει να παίζει το χρονοδιάγραμμα της MovieClip το κουμπί τοποθετείται in. Το ίδιο ισχύει και για το χρονοδιάγραμμα ρίζα από τη ρίζα θεωρείται Movie Clip επίσης. (Σημείωση: Η μέθοδος αυτή της κωδικοποίησης αποδοκιμάστηκε και δεν θα πρέπει πλέον να χρησιμοποιείται).

```
on(release)
```

```
{  
    play();  
}
```

Αυτό είναι πιο προχωρημένο παράδειγμα, δημιουργεί έναν πίνακα που περιέχει τους αριθμούς και τα string, και αποδίδει έναν αριθμό σε μια μεταβλητή που ονομάζεται num και ένα string σε μια μεταβλητή που ονομάζεται str. πρωτότυπο, χρησιμοποιώντας τις λειτουργίες και τα καθήκοντα αναδρομής. Στη συνέχεια, χρησιμοποιώντας το **Movie Clip** API, είναι ένα πεδίο κειμένου που εμφανίζεται στην οθόνη σας, με τιμές που η μεταβλητή είναι γραμμένη.

```
Array.prototype.pickNumber = function():Number  
{  
    var rand:Number = Math.floor(Math.random() * this.length);  
    return (typeof (this[rand]) == "number") ? this[rand] :  
this.pickNumber();  
};  
  
Array.prototype.pickString = function():String  
{  
    var rand:Number = Math.floor(Math.random() * this.length);  
    return (typeof (this[rand]) == "string") ? this[rand] :  
this.pickString();  
};  
  
var myArray:Array = new Array("Hello", "ActionScript", 3, 7, 11,  
"Flash");  
  
var num:Number = myArray.pickNumber();  
  
var str:String = myArray.pickString();  
  
this.createTextField("txt", 1, 10, 10, 530, 390);
```

```
txt.text = "Array = " + myArray + "\nRandom Number = " + num +
"\nRandom String = " + str;
```

Array και data Provider παράδειγμα :

```
var aData:Array = [{name: "J. Bell", age: "55"}, {name: "B. Longman",
age: "21"}];

dataGrid.dataProvider = aData;
```

Action Script 3.0 παραδείγματα

Αυτό το παράδειγμα "Hello World" χρησιμοποιεί Action Script 3.0:

```
package {

    import flash.display.Sprite;

    import flash.text.TextField;

    public class HelloWorld extends Sprite {

        public function HelloWorld() {

            var txt:TextField = new TextField();

            txt.text = "Hello World!";

            addChild(txt);

        }

    }

}
```

1.13 Δομές Δεδομένων

1.13.1 Τύποι δεδομένων

Η Action Script κατά κύριο λόγο αποτελείται από "θεμελιώδη" ή "απλούς" τύπους δεδομένων που χρησιμοποιούνται για τη δημιουργία άλλων τύπων δεδομένων. Τα στοιχεία αυτά, είδη είναι πολύ παρόμοια με Java τύπους δεδομένων. Δεδομένου ότι Action Script 3.0 ήταν ένα πλήρες ξαναγράφηκε σε Action Script 2.0, οι τύποι στοιχείων και οι κληρονομίες τους έχουν αλλάξει.

Action Script 2.0 υψηλού επιπέδου τύποι δεδομένων

- **String** - ένας κατάλογος χαρακτήρων όπως "Hello World"
- **Αριθμός** - Είναι οποιαδήποτε αριθμητική αξία
- **Boolean** - μια απλή δυαδική αποθήκευση που μπορεί μόνο να είναι true ή false.
- **Αντικείμενο** - το αντικείμενο είναι το στοιχείο που δακτυλογραφεί όλους τους σύνθετους τύπους στοιχείων που κληρονομεί. Επιτρέπει την ομαδοποίηση των μεθόδων, των λειτουργιών, των παραμέτρων, και άλλων αντικειμένων.

Action Script 2.0 σύνθετη τύποι δεδομένων

Υπάρχουν και επιπρόσθετοι "σύνθετοι" τύποι δεδομένων. Υπάρχουν περισσότεροι επεξεργαστές και μνήμες και αποτελούνται από πολύ "απλούς" τύπους στοιχείων. Για AS2, ορισμένα από τα δεδομένα αυτά είδη είναι τα εξής:

- **MovieClip** – Action Script μια δημιουργία που επιτρέπει την εύκολη χρήση στα ορατά αντικείμενα.
- **TextField** - Μία απλή δυναμική είσοδο ή πεδίο κειμένου. Κληρονομεί τον Movieclip τύπο.
- **Button** - Ένα απλό κουμπί με 4 καρτέ (μέλη): Πάνω, Πάνω, κάτω και Hit. Κληρονομεί το MovieClip τύπου.
- **Ημερομηνία** - Επιτρέπει την πρόσβαση σε πληροφορίες για ένα συγκεκριμένο χρονικό σημείο.
- **Array** - Επιτρέπει την γραμμική αποθήκευση των δεδομένων.

- **XML** – Είναι ένα αντικείμενο XML
- **XMLNode** – Είναι ένας κόμβος XML
- **LoadVars** - ένα φορτίο μεταβλητές αντικειμένου επιτρέπει την αποθήκευση και αποστολή του HTTP GET και HTTP POST μεταβλητές
- **Sound**
- **NetStream**
- **NetConnection**
- **MovieClipLoader**
- **EventListener**

Action Script 3.0 υψηλού επιπέδου τύποι δεδομένων

- **Boolean**- Ο τύπος δεδομένων έχει μόνο δύο δυνατές τιμές: αληθής και ψευδής ή 1 και 0, δεν ισχύουν άλλες τιμές.
- **Int** - Ο int τύπος δεδομένων είναι ένα 32-bit ακέραιος μεταξύ - 2.147.483.648 και 2.147,483647 εκατομμύρια.
- **Null** –Ουδέτερος τύπος δεδομένων περιέχει μία μόνο τιμή null. Αυτή είναι η προεπιλεγμένη τιμή για τα δεδομένα τύπου String και όλες τις κατηγορίες που ορίζουν πολύπλοκους τύπους δεδομένων , συμπεριλαμβανομένων των αντικειμένων της κατηγορίας.
- **Αριθμός** - Ο αριθμός δεδομένων μπορεί να αντιπροσωπεύει ακέραιους , χωρίς υπογραφή, και αριθμούς κινητής υποδιαστολής. Ο Αριθμός των δεδομένων χρησιμοποιεί το 64-bit διπλής ακρίβειας μορφή, όπως ορίζεται από το πρότυπο IEEE για Binary Floating-Point Arithmetic (IEEE-754).
- **String** -ο τύπος στοιχείων String αντιπροσωπεύει μια ακολουθία 16-bit χαρακτήρων. Οι σειρές αποθηκεύονται εσωτερικά ως χαρακτήρες Unicode, χρησιμοποιώντας το σχήμα utf-16. Οι προηγούμενες εκδόσεις του flash χρησιμοποίησαν το σχήμα utf-8.
- **uint** - ο τύπος στοιχείων uint (ανυπόγραφος ακέραιος αριθμός) είναι ένας 32-bit ανυπόγραφος ακέραιος αριθμός μεταξύ 0 και 4.294.967.295.
- **Void** - ο void τύπος στοιχείων περιέχει μόνο μια αξία, απροσδιόριστη. Στις προηγούμενες εκδόσεις Action Script, απροσδιόριστες ήταν η προκαθορισμένη αξία για τις περιπτώσεις της κατηγορίας αντικειμένου. Σε

Action Script 3.0, η προκαθορισμένη αξία για τις περιπτώσεις αντικειμένου είναι μηδενική.

Action Script 3 σύνθετοι τύποι δεδομένων

- **Αντικείμενο** - ο τύπος των στοιχείων αντικειμένου καθορίζεται από την κατηγορία αντικειμένου. Η κατηγορία αντικειμένου χρησιμεύει ως η κατηγορία βάσεων για όλους τους ορισμούς κατηγορίας σε Action Script.
- **Array** - Περιέχει έναν κατάλογο στοιχείων. Αν και Action Script 3 είναι μια έντονα-δακτυλογραφημένη γλώσσα, δεν υποστηρίζει τους δακτυλογραφημένους Arrays. Κατά συνέπεια το περιεχόμενο ενός Array μπορεί να είναι οποιουδήποτε τύπου.
- **MovieClip** - Animated movie clip αντικείμενο επίδειξης ένας απόγονος (με τις δευτερεύουσες τροποποιήσεις) της κύριας υπόδειξης ως προς Flash timeline.
- **Bitmap** - Ένα μη animated αντικείμενο επίδειξης bitmap.
- **Shape** - Ένα μη animated ανυσματικό αντικείμενο μορφής.
- **Byte Array** - Περιέχει ένα πίνακα από δυαδικά byte δεδομένων.
- **TextField** - Ένα δυναμικό, προαιρετικά διαδραστικό κείμενο πεδίου αντικείμενου.
- **SimpleButton** - Ένας απλός διαλογικός τύπος κουμπιών που υποστηρίζει " up ", " over», και " down" με μια αυθαίρετη περιοχή χτυπήματος.
- **Ημερομηνία** - Ένα αντικείμενο ημερομηνίας που περιέχει την τρέχοντες ημερομηνία/το χρόνο συστημάτων.
- **Error** - Ένα γενικό σφάλμα αντικειμένου που επιτρέπει την εκτέλεση της αναφοράς σφάλματος όταν ρίχνονται ως εξαίρεση.
- **Function** - Η βασική κατηγορία για Flash μέθοδο ορισμού.
- **RegExp** - Μία κανονική έκφραση αντικειμένου για strings.
- **Video** - Μία τηλεοπτική υποστήριξη αντικειμένου αναπαραγωγής ήχου άμεση (προοδευτικός μεταφορτώστε) ή μεταφορές ροής (RTMP). Από την έκδοση Flash Player version 9.0.115.0 το ανοικτό υψηλής ευκρίνειας

τηλεοπτικό σχήμα h.264/MP4 υποστηρίζεται επίσης κατά μήκος του δευτερεύοντος τυποποιημένου τηλεοπτικού (FLV) περιεχομένου Flash.

- **XML** - Ένα αναθεωρημένο αντικείμενο XML βασισμένο στα E4C πρότυπα οι κόμβοι και οι ιδιότητες προσεγγίζονται διαφορετικά από Action Script 2.0 στο αντικείμενο (μια κατηγορία κληρονομιών που ονομάζεται XMLDocument παρέχεται για προς τα πίσω τη συμβατότητα).
- **XMLList** – Ένας πίνακας βασισμένος στο αντικείμενο για τις διάφορες ικανοποιημένες ματιές στην κατηγορία XML.

1.13.2 Χρήση των τύπων δεδομένων

Η βασική σύνταξη είναι:

```
var yourVariableName:YourVariableType = new YourVariableType (Param1, Param2, ..., ParamN);
```

Έτσι προκειμένου να γίνει ένα άδειο αντικείμενο:

```
var myObject:Object = new Object();
```

Some types are automatically put in place:

```
var myString:String = "Hello Wikipedia!"; // Αυτό θα ορίζεται αυτόματα τη μεταβλητή ως string.
```

```
var myNumber:Number = 5; // Αυτό θα κάνει το ίδιο για έναν αριθμό.
```

```
var myObject:Object = {Param1:"Hi!", Param2:76}; // Αυτό δημιουργεί ένα αντικείμενο με δύο μεταβλητές.
```

```
// Param1 είναι ένα string με τα στοιχεία του "Hi!",
```

```
// Param2 και είναι ένας αριθμός με τα δεδομένα 76.
```

```
var myArray:Array = [5, "Hello!", {a:5, b:7}] // Αυτή είναι η σύνταξη για αυτόματη δημιουργία ενός σχηματισμού.
```

```
// Δημιουργεί έναν πίνακα με 3 μεταβλητές.
```

```
// Η πρώτη (0) είναι ένας αριθμός με την τιμή 5,
```

```
// το δεύτερο (1) είναι ένα string με την αξία των "Hello!",  
// και το τρίτο (2) είναι ένα αντικείμενο με (α: 5, β: 7).
```

Αντίθετα από τις περισσότερες αντικειμενοστραφείς γλώσσες, η Action Script δεν κάνει καμία διάκριση μεταξύ των πρωτόγονων τύπων και των τύπων αναφοράς. Στην Action Script όλες οι μεταβλητές είναι τύποι αναφοράς.

Σε μερικούς τύπους στοιχείων μπορούν να οριστούν οι τιμές με τα literals:

```
var item1:String="ABC";  
var item2:Boolean=true;  
var item3:Number=12  
var item4:Array=["a", "b", "c"];  
var item5:XML = <node><child/></node>; // Σημειώστε ότι η πρωτόγονη XML δεν  
είναι εισηγμένη
```

Μια αναφορά σε Action Script είναι ένας δείκτης σε μια περίπτωση μιας κατηγορίας. Αυτό δεν δημιουργεί ένα αντίγραφο αλλά έχει πρόσβαση στο ίδιο διάστημα μνήμης. Όλα τα αντικείμενα σε Action Script προσεγγίζονται ως αναφορές αντί της αντιγραφής.

```
var item1:XML=new XML("<node><child/></node>");  
var item2:XML=item1;  
item2.firstChild.attributes.value=13;  
// item1 τώρα ισούται item2 από item2 απλά σημεία σε ότι item1 επισημαίνει.  
// Και οι δύο είναι τώρα:  
//<node><child value="13"/></node>
```

Η απομάκρυνση των αντικειμένων και δεδομένων γίνεται από το Flash Player συλλέκτη σκουπιδιών που ελέγχει για τυχόν υπάρχουσες αναφορές στην Flash μνήμη. Αν δεν υπάρχουν (δεν υπάρχει άλλο αντικείμενο που χρησιμοποιούν είναι το ορφανό αντικείμενο), αυτό αφαιρείται από τη μνήμη. Μόνο οι αναφορές μπορούν να εξαλειφθούν με τη χρήση του "διαγραφή" λέξεων-κλειδιών. Για το λόγω αυτό, στη διαχείριση μνήμης Action Script είναι δύσκολη και απαιτεί προσεκτική εφαρμογή αναπτυξιακού σχεδιασμού.

```
var item1:XML=new XML("<node><child/></node>");
```

```
delete item1;
```

```
// Εάν δεν υπάρχει άλλη αναφορά σε item1 δεν υπάρχει πουθενά αλλού στην αίτηση,
```

```
// θα μετατεθεί στα απορρίματα collector' επόμενο πέρασμα του
```

1.14 Ανταγωνισμός

1.14.1 Format and plug-in

Έναντι άλλων plug-ins όπως η Java, το Acrobat Reader, το QuickTime, ή Windows Media Player. Το Flash Player έχει ένα μικρό μέγεθος εγκατάστασης, μεταφορτώνοντας γρήγορα τον χρόνο, και τον γρήγορο χρόνο έναρξης. Ωστόσο, πρέπει να ληφθεί μέριμνα για την ανίχνευση και την παγίωση του Flash Player στον (X) HTML σε ένα W3C συμβατό τρόπο. [Σημείο αναφοράς που απαιτείται] Μια απλή και ευρεία χρήση είναι ο εναλλακτικός τρόπος που ακολουθεί:

```
<object data="movie.swf" type="application/x-shockwave-flash"
width="500" height="500">
  <param name="movie" value="movie.swf" />
</object>
```

Περισσότερες πληροφορίες για το πώς να εντοπίσετε και να ενσωματώσετε Flash αντικείμενα σε μια W3C με συμβατό τρόπο παρέχεται στο **xSWF** περιγραφή. Η χρήση της ανυσματικής γραφικής παράστασης που συνδυάζεται με τον κώδικα προγράμματος επιτρέπει στα αρχεία του Flash για να είναι μικρότερα, ή στις ροές να χρησιμοποιήσει το λιγότερο εύρος ζώνης, από τα αντίστοιχα δυαδικά αρχεία εικόνας ή τα βιντεοκλίπ. Για το περιεχόμενο με ένα ενιαίο σχήμα (όπως ακριβώς το κείμενο, το βίντεο ή ο ήχος) άλλες εναλλακτικές λύσεις μπορούν να παρέχουν την καλύτερη απόδοση και να καταναλώσουν τη λιγότερη **cpu** από τον αντίστοιχο **Flash movie**, παραδείγματος χάριν κατά την χρησιμοποίηση της διαφάνειας ή την κατασκευή της μεγάλης οθόνης ενημερώνει όπως φωτογραφική ή κείμενο εξασθένιση.

Επιπρόσθετα σε μια μηχανή ανυσματικής- απόδοσης, ο Flash Player περιλαμβάνει ένα μηχανήμα εικονικής πραγματικότητας που καλείται μηχανήμα εικονικής πραγματικότητας **Action Script (AVM)** για την αλληλεπίδραση σεναριογράφων στο χρόνο εκτέλεσης, υποστήριξη για τον τηλεοπτικό, mp3-βασισμένο ήχο, και τη γραφική παράσταση δυαδικών αρχείων εικόνας. Από το Flash Player 8, προσφέρεται στο βίντεο δύο **codec's**: On2 τεχνολογίες VP6 και **Sorenson Spark**, και υποστήριξη χρόνου εκτέλεσης για JPEG, προοδευτικό JPEG, το PNG, και το GIF. Στην επόμενη έκδοση, το Flash είναι επικριμένο να χρησιμοποιήσει έναν πάνω στην ώρα μεταγλωττιστή για τη μηχανή Action Script.

1.14.2 Το Flash στην αγορά

Το Flash ως σχήμα έχει γίνει πολύ διαδεδομένο στην αγορά υπολογιστών γραφείου και έχει δημιουργήσει μια κυριαρχία αγοράς. Το Adobe υποστηρίζει ότι 98 τοις

εκατό των χρηστών αμερικανικού Ιστού και 99.3 τοις εκατό όλων των χρηστών υπολογιστών γραφείου Διαδικτύου εγκαθιστούν το Flash Player, με 45%-56% (ανάλογα με την περιοχή) που έχει την πιο πρόσφατη έκδοση. Οι αριθμοί ποικίλλουν ανάλογα με το **demographics** σχεδίου και έρευνας ανίχνευσης.

Το **Adobe Flash Player** υπάρχει για ποικίλες συστήματα και συσκευές: **Windows, MAC OS 9/X, Linux (μόνο 32-bit x86), Solaris, HP-UX, PC τσεπών, OS/2, QNX, Symbian, Palm OS, BeOS, και IRIX**). Μια αξιοσημείωτη εξαίρεση είναι υποστήριξη για τα **64-bit** λειτουργικά συστήματα (δείτε την κριτική). Για τη συμβατότητα με τις συσκευές (ενσωματωμένα συστήματα), δείτε **Macromedia Flash Lite**.

1.14.3 Κλασικές εναλλακτικές

Το **W3C**, τα πρότυπα **SVG** και **SMIL** είναι οι ποιώ στενοί ανταγωνιστές του Flash. Το Adobe που χρησιμοποιείται για να αναπτύξει και να διανείμει το '**Adobe SVG Viewer**' η σύνδεση πελατών για το **MS Internet Explorer**, αλλά έχει αναγγείλει πρόσφατα την αναστολή της. Έχει σημειωθεί από τους σχολιαστές βιομηχανίας ότι αυτό δεν ήταν πιθανώς καμία σύμπτωση σε μία εποχή που κινήθηκε η Adobe από τον ανταγωνισμό με Macromedia's Flash, στην ιδιοκτησία της ίδιας της τεχνολογίας. Εν τω μεταξύ, έχει υποστηρίξει SVG δεδομένου ότι η έκδοση 8 και το σαφάρι έχουν από την έκδοση 3, και **Firefox** η ενσωματωμένη υποστήριξη του για SVG συνεχίζει να αυξάνεται.

Η UIRA ήταν ένα πρόγραμμα ελεύθερου λογισμικού που σκόπευε να γίνει μια πλήρης αντικατάσταση για τη Adobe Flash. Το πρόγραμμα κατέρρευσε στα μέσα του 2007, αν και οι άνθρωποι συζητούν τώρα να αναβιώσουν ή να το συνεχίσουν.

1.15 Τρίτου επιπέδου υλοποίηση

1.15.1 Λεπτομερής παρουσίαση

Τον Οκτωβρίου του 1998 , η Macromedia αποκάλυψε την έκδοση Flash 3 προδιαγραφή στον κόσμο σε κάθε ιστοσελίδα. Έκανε αυτό σε απάντηση για πολλά νέα και συχνά **semi-open** σχήματα που ανταγωνίζονται με SWF, όπως **Xara's Flare** και **Sharp' Extended Vector Animation formats**. Οι διάφοροι υπεύθυνοι για την ανάπτυξη δημιούργησαν γρήγορα τη **C** βιβλιοθήκη για την παραγωγή **SWF**. Τον Φεβρουάριο του 1999, η επιχείρηση εισήγαγε **Morphink 99**, το πρώτο πρόγραμμα τρίτων για να δημιουργήσουν τα αρχεία **SWF**. Το Macromedia μίσθωσε επίσης το **Middlesoft** για να δημιουργήσει ένα ελεύθερα διαθέσιμο developers' εξάρτημα για τις εκδόσεις μορφής αρχείου SWF 3 έως 5.

Το Macromedia έκανε τις προδιαγραφές αρχείων Flash για τις εκδόσεις 6 και ποιο πρόσφατα διαθέσιμες μόνο στο πλαίσιο μιας συμφωνίας μη-κοινοποίησης, αλλά είναι ευρέως διαθέσιμο από τις διάφορες περιοχές. Τον Απρίλιο του 2006, η προδιαγραφή της μορφής αρχείου **Flash SWF** απελευθερώθηκε με τις λεπτομέρειες στο τότε νεότερο σχήμα έκδοσης (**Flash 8**). Αν και ακόμα λείπουν οι συγκεκριμένες πληροφορίες για τα ενσωματωμένα τηλεοπτικά σχήματα συμπίεσης (**On2, Sorenson Spark**, κ.λπ.), αυτή η νέα τεκμηρίωση κάλυψε όλα τα νέα χαρακτηριστικά γνωρίσματα πρόσφερε στο **Flash v8** συμπεριλαμβανομένων των νέων Action Script εντολών, εκφραστικοί έλεγχοι φίλτρων, κτλ. Το έγγραφο των προδιαγραφών μορφής αρχείου προσφέρεται μόνο στους υπεύθυνους για την ανάπτυξη που συμφωνούν με μια έγκριση αδειας που επιτρέπει να χρησιμοποιήσει τις προδιαγραφές για να αναπτύξει μόνο τα προγράμματα που μπορούν να εξαγάγουν στη μορφή αρχείου Flash. Η άδεια απαγορεύει τη χρήση των προδιαγραφών για να δημιουργήσει τα προγράμματα που μπορούν να χρησιμοποιηθούν για την αναπαραγωγή ήχου των αρχείων Flash. Η Flash 9 προδιαγραφή παρέχεται κάτω από τους παρόμοιους περιορισμούς. Τον Μάιο του 2008, η **Adobe** προώθησε το ανοικτό πρόγραμμα οθόνης (σύνδεση Adobe), το οποίο κατέστησε την προδιαγραφή SWF διαθέσιμη χωρίς περιορισμούς.

1.15.2 Playback

Δεδομένου ότι τα αρχεία του Flash δεν εξαρτώνται από ένα ανοικτό πρότυπο όπως SVG, αυτό μειώνει το κίνητρο για το μη εμπορικό λογισμικό για να υποστηρίξει το σχήμα, [παραπομπή που απαιτείται] αν και υπάρχουν διάφορα εργαλεία τρίτων

που χρησιμοποιούν και παράγουν τη μορφή αρχείου SWF. Το **IrfanView** είναι ικανό να παίζει μορφή αρχείων SWF. Υπάρχει μία μεγάλη και δονούμενη ανοικτή κοινότητα πηγής. Το Flash Player δεν μπορεί να στείλει ως τμήμα μιας καθαρής ανοικτή πηγή, ή του απολύτως ελεύθερου λειτουργικού συστήματος, δεδομένου ότι η διανομή της υποβάλλεται συνδεδεμένη στο πρόγραμμα χορήγησης αδειών Macromedia και στην έγκριση.

Δεν υπάρχει, όπως τα τέλη 2007, καμία πλήρης αντικατάσταση ελεύθερου λογισμικού που προσφέρει όλη τη λειτουργία της πιο πρόσφατης έκδοσης του **Adobe Flash Player**.

Το **Gnash** είναι ένα ενεργό πρόγραμμα που στοχεύει να δημιουργήσει έναν ελεύθερο φορέα και μια μηχανή αναζήτησης plug-in για τη μορφή αρχείου **Adobe Flash** και να παρέχει έτσι μια ελεύθερη εναλλακτική λύση στο Adobe Flash Player με άδεια ευρέος κοινού **GNU**. Με τις πιθανές ανησυχίες διπλωμάτων ευρεσιτεχνίας λόγω της ιδιόκτητης φύσης των σχετικών αρχείων, των υποστηρίξεων Gnash τα περισσότερα SWF v7 χαρακτηριστικά γνωρίσματα και κάποιο SWF v8 και v9. Τρεξίματα Gnash στα παράθυρα, Linux και άλλα λειτουργικά συστήματα στις 32-bit, 64-bit και άλλες αρχιτεκτονικές. Το **Swfdec** είναι ένας άλλος Flash Player ανοικτής-πηγής διαθέσιμος για **Linux και FreeBSD**. Δείτε επίσης **SWFOpener**.

1.15.3 Authoring

Προγράμματα του Open Source όπως τον **Ajax Animator**, και (τώρα defunct) στόχος UIRA να δημιουργηθεί ένα περιβάλλον ανάπτυξης Flash, πλήρες με ένα γραφικό περιβάλλον χρηστών. Εναλλακτικά, προγράμματα όπως το **swfmill**, **SWFTools**, και **MTASC** παρέχουν τα εργαλεία για να δημιουργήσουν τα αρχεία SWF, αλλά κάνουν έτσι και την σύνταξη του κειμένου, **action script** ή των αρχείων **XML** στο **Flash animations**. Είναι επίσης δυνατό να δημιουργηθούν τα αρχεία SWF που χρησιμοποιούν programmaticaly τη βιβλιοθήκη Ming, η οποία έχει τις διεπαφές για το **C**, **PHP**, **C++**, **Perl**, **Python**, και **Ruby**. Η **HaXe** είναι μια ανοικτή πηγή, υψηλού επιπέδου αντικειμενοστραφής γλώσσα προγραμματισμού που συνδέεται προς την Ιστό-ικανοποιημένη δημιουργία που μπορεί να συντάξει τα αρχεία Flash.

Πολλοί υπεύθυνοι για την ανάπτυξη διανεμημένων λογισμικών παρήγαγαν τα εργαλεία Flash και τα πούλησαν για κατώτερο US\$50 μεταξύ 2000 και 2002. Το 2003 ο ανταγωνισμός και η εμφάνιση των ελεύθερων εργαλείων Flash είχαν οδηγήσει πολλούς εργαλείο-κατασκευαστές από την αγορά, που επιτρέπει στους υπόλοιπους υπεύθυνους την ανάπτυξη για να αυξήσουν τις τιμές τους, αν και πολλά από τα

προϊόντα κόστισαν ακόμα λιγότερο από US\$100 και την υποστήριξη Action Script. Όσον αφορά στα ανοικτά εργαλεία πηγής, KToon μπορεί να εκδώσει τα διανύσματα και να παραγάγει SWF, αλλά η διεπαφή της είναι πολύ διαφορετική από Macromedia. Ο **Toon Boom Technologies** πουλάει επίσης το παραδοσιακό εργαλείο animation, βασισμένο στη Flash - Toon- Army.

Άλλα εργαλεία στρέφονται στη δημιουργία των συγκεκριμένων τύπων περιεχομένου Flash. Το στούντιο Anime είναι ένα 2D λογισμικό animation που ειδικεύεται για τον animation χαρακτήρα που δημιουργεί τα αρχεία SWF

Οι χρήστες που δεν είναι προγραμματιστές ή σχεδιαστές Ιστού θα βρουν επίσης τα σε απευθείας σύνδεση εργαλεία που επιτρέπουν να χτίσουν πλήρεις Flash-based ιστοσελίδες. Μία από τις πιο παλαιότερες διαθέσιμες υπηρεσίες το (1998) είναι **FlashToGo**. Τέτοιες επιχειρήσεις παρέχουν μια ευρεία ποικιλία των προ-χτισμένων προτύπων (πρότυπα) που συνδέονται σε ένα ικανοποιημένο σύστημα διαχείρισης που εξουσιοδοτεί τους χρήστες για να χτίσει εύκολα, να εκδώσει και να δημοσιεύσει τις ιστοσελίδες τους.

Η Adobe έγραψε ένα πακέτο λογισμικού αποκαλούμενο **Adobe LiveMotion**, με σκοπό να δημιουργήσει το διαλογικό περιεχόμενο animation και να το εξαγάγει σε ποικίλα σχήματα, συμπεριλαμβανομένου SWF. Το **LiveMotion** πέρασε από δύο σημαντικές απελευθερώσεις, αλλά απέτυχε να κερδίσει οποιαδήποτε ξεχωριστή βάση χρηστών.

Το Φεβρουαρίου του 2003, η Macromedia αγόρασε **Presedia**, το οποίο είχε αναπτύξει ένα εργαλείο δημιουργίας Flash που μετέτρεπε αυτόματα τα αρχεία του PowerPoint στο Flash. Η Macromedia απελευθέρωσε στη συνέχεια το νέο προϊόν ως αεράκι, το οποίο περιέλαβε πολλές νέες αυξήσεις. Επιπλέον, (από την έκδοση 2) Apple' το βασικό λογισμικό παρουσίασης επιτρέπει επίσης στους χρήστες για να δημιουργήσει τις διαλογικές παρουσιάσεις και την εξαγωγή σε SWF.

1.16 Σχετικά αρχεία τύποι και επεκτάσεις

Ext.	Explanation
.swf	Τα .Swf αρχεία που συμπληρώνονται, συντάσσονται και δημοσιεύονται τα οποία δεν μπορούν να εκδοθούν με το Adobe Flash. Εντούτοις, υπάρχουν πολλά ' .swf decompilers'. Η προσπάθεια να εισαχθούν .swf αρχεία που χρησιμοποιούν το Flash επιτρέπει στο να ανακτήσει μερικά προτερήματα από .swf, αλλά όχι όλων.
.fla	Τα .fla αρχεία περιέχουν το υλικό πηγής για την εφαρμογή Flash. Το λογισμικό δημιουργίας Flash μπορεί να εκδώσει τα αρχεία FLA και να τα συντάξει .swf αρχεία.
.xfl	.xfl τα αρχεία είναι xML βασισμένα αρχεία προγράμματος που είναι ισοδύναμα με το δυαδικό .fla σχήμα. Το λογισμικό δημιουργίας Flash θα χρησιμοποιήσει XFL ως ανταλλαγής στη Flash CS4. Θα εισάγει αρχεία XFL που εξάγονται από In Design.
.as	Τα .as αρχεία περιέχουν το κωδικό πηγής Action Script στα απλά αρχεία πηγής. Τα αρχεία FLA μπορούν επίσης να περιέχουν τον κώδικα Action script άμεσα, αλλά τα χωριστά εξωτερικά .as αρχεία προκύπτουν συχνά για διαρθρωτικούς λόγους, ή για να εκθέσουν τον κώδικα οι εφαρμογές. Χρησιμοποιούν μερικές φορές την επέκταση .action script
.swd	Τα .swd αρχεία είναι προσωρινά αρχεία διόρθωσης που χρησιμοποιούνται κατά τη διάρκεια της ανάπτυξης Flash. Μόλις τελειώσει ένα πρόγραμμα Flash αυτά τα αρχεία δεν είναι απαραίτητα και μπορούν να αφαιρεθούν.
.asc	Τα .asc αρχεία περιέχουν μια δευτερεύον Action Script, η οποία χρησιμοποιείται για να αναπτύξει τον αποδοτικό και εύκαμπτο εξυπηρετητή επικοινωνίας MX Flash Macromedia κεντρικών υπολογιστών πελατών
.flv	Τα .flv αρχεία είναι τηλεοπτικά αρχεία flash, όπως δημιουργούνται από τη Adobe Flash, ffmpeg, τη συμπίεση Sorenson, ή On2 Flix.

.f4v Τα .f4v αρχεία είναι τυποποιημένα mp4 αρχεία που μπορούν να παιχτούν πίσω από το Flash Player 9 αναπροσαρμογή 3 και άνω.

.f4p Τα .f4p αρχεία είναι mp4 αρχεία με την ψηφιακή διαχείριση δικαιωμάτων.

.f4a Τα .f4a αρχεία είναι mp4 αρχεία που περιέχουν μόνο τα ακουστική ροή.

.f4b Τα .f4b αρχεία είναι mp4 ακουστικά αρχεία βιβλίων.

.swc Τα .swc αρχεία χρησιμοποιούνται για τη διανομή των συστατικών ,περιέχουν έναν συνταγμένο συνδετήρα, το component' αρχείο κατηγορίας του Action Script, και άλλα αρχεία που περιγράφουν το συστατικό.

.jsfl Τα .jsfl αρχεία χρησιμοποιούνται για να προσθέσουν τη λειτουργία στο περιβάλλον δημιουργίας Flash ,περιέχουν τον κώδικα Javascript και έχουν πρόσβαση στη Flash Javascript API.

.swt Τα .swt αρχεία είναι ' templated' μορφές .swf αρχείων, που χρησιμοποιούνται από Macromedia Generator

.flp Τα .flp αρχεία είναι αρχεία XML που χρησιμοποιούνται για να παραπέμψουν όλο το περιλαμβανόμενο πρόγραμμα εγγράφων. Τα προγράμματα Flash επιτρέπουν στο χρήστη να συγκεντρώσουν τα πολλαπλάσια σχετικά αρχεία για να βοηθήσουν στην οργάνωση προγράμματος Flash, σύνταξη και να χτίσουν.

.spl Τα .spl αρχεία είναι έγγραφα Future Splash.

.aso Τα .aso αρχεία είναι αρχεία κρυπτής που χρησιμοποιούνται κατά τη διάρκεια της ανάπτυξης Flash, που περιέχει το συνταγμένο κώδικα ψηφιακών λέξεων Action Script. Ένα αρχείο ASO αναδημιουργείται όταν ανιχνεύεται μια αλλαγή στα αντίστοιχα αρχεία κατηγορίας της. Περιστασιακά το Flash IDE δεν αναγνωρίζει ότι ένας recompile είναι απαραίτητος, και αυτά τα αρχεία κρυπτής πρέπει να διαγραφούν με το χέρι. Βρίσκονται %USERPROFILE%\Local Settings\Application Data\Macromedia\Flex8\en\Configuration\Classes\aso on Win32 / Flex8.

.lmv Αυτά τα αρχεία δημιουργούνται από το πρόγραμμα δωρεάν λογισμικού αποκαλούμενο liveswif. Χρησιμοποιούνται για να σώσουν το animation σε ένα editable αρχείο, αλλά μπορούν επίσης να μετατραπούν σε ένα .swf αρχείο για να παραγάγουν σε απευθείας σύνδεση περιεχόμενο για τον Ιστό. Αυτό το αρχείο δεν έχει καμία σχέση με το αρχείο Fla Adobe Flash, με τη μόνη ομοιότητα που είναι ότι και οι δύο φυλάσσουν τα editable στοιχεία που μπορούν να μετατραπούν σε ένα αρχείο swf.

1.17 Video στις σελίδες του web

Το Flash μπορεί να χρησιμοποιηθεί για να ενσωματώσει το βίντεο σε μια ιστοσελίδα, ένα χαρακτηριστικό γνώρισμα διαθέσιμο από την έκδοση Flash Player 6. Η τεχνική είναι να δημιουργήσει ένα αρχείο Flash (.swf) που ενεργεί ως φορέας για το τηλεοπτικό αρχείο. Αυτό είναι η βάση για πολλά δημοφιλείς video sites, συμπεριλαμβανομένου YouTube και του Google Video. Το πραγματικό βίντεο αρχείο είναι είτε ένα αρχείο FLV είτε H.264, και τα δύο μπορούν εύκολα να παιχτούν από το γενικό λογισμικό video player. Εντούτοις, παίρνει τις μηχανές αναζήτησης για να επιδειχθεί το βίντεο, είναι ακόμα ένα συγκεκριμένο ζήτημα πλατφορμών λόγω της έλλειψης ενός κοινού βίντεο σχήματος, και το θέμα προτύπων Ιστού για το βίντεο είναι μια έντονη αντιπαράθεση. Η χρησιμοποίηση του Flash λύνει εν μέρει το πρόβλημα λόγω της ευρείας διανομής του Flash Player, αλλά δεδομένου ότι αυτό είναι ιδιόκτητης τεχνολογίας για την οποία δεν υπάρχει καμία πραγματική εναλλακτική λύση, καθιστά τα μέσα εμφανώς δύσκολα να έχουν πρόσβαση για non-users του Flash Player (ιδιαίτερα εάν η θέση του βίντεο κινείται από την πηγή HTML). Ένα άλλο μειονέκτημα της επίδειξης του βίντεο μέσω του Flash, είναι ότι ένας Flash Player πρέπει να είναι σε θέση να ζωντανέψει πάνω από την τηλεοπτική απόδοση, η οποία κάνει επιταχυνόμενο το υλικό βίντεο καθιστώντας τουλάχιστον όχι τόσο απλό όσο με έναν κατασκευασμένο επί τούτου φορέα πολυμέσων. Πράγματι, ο Adobe Flash Player είχε όχι πάντα την καλή εκτέλεση σε όλες τις πλατφόρμες (δείτε την κριτική), και δεν είναι ακόμα ασυνήθιστο για άλλους φορείς πολυμέσων να παίζουν καλά όπου το Flash Player ρίχνει τα πλαίσια και πηδά τον ήχο.

1.17.1 Flash Video

Το Flash βίντεο (.flv αρχεία) είναι ένα σχήμα εμπορευματοκιβωτίων, που σημαίνει ότι δεν είναι ένα τηλεοπτικό σχήμα στον εαυτό του, αλλά μπορεί να περιέχει άλλα σχήματα. Το βίντεο στο Flash κωδικοποιείται σε H.263, και αρχικά με τον Flash player 8, μπορεί εναλλακτικά να κωδικοποιηθεί VP6. Ο ήχος είναι MP3. Η χρήση

VP6 είναι κοινή σε πολλές επιχειρήσεις, λόγω των μεγάλων ποσοστών υιοθέτησης Flash player 8 και Flash player 9.

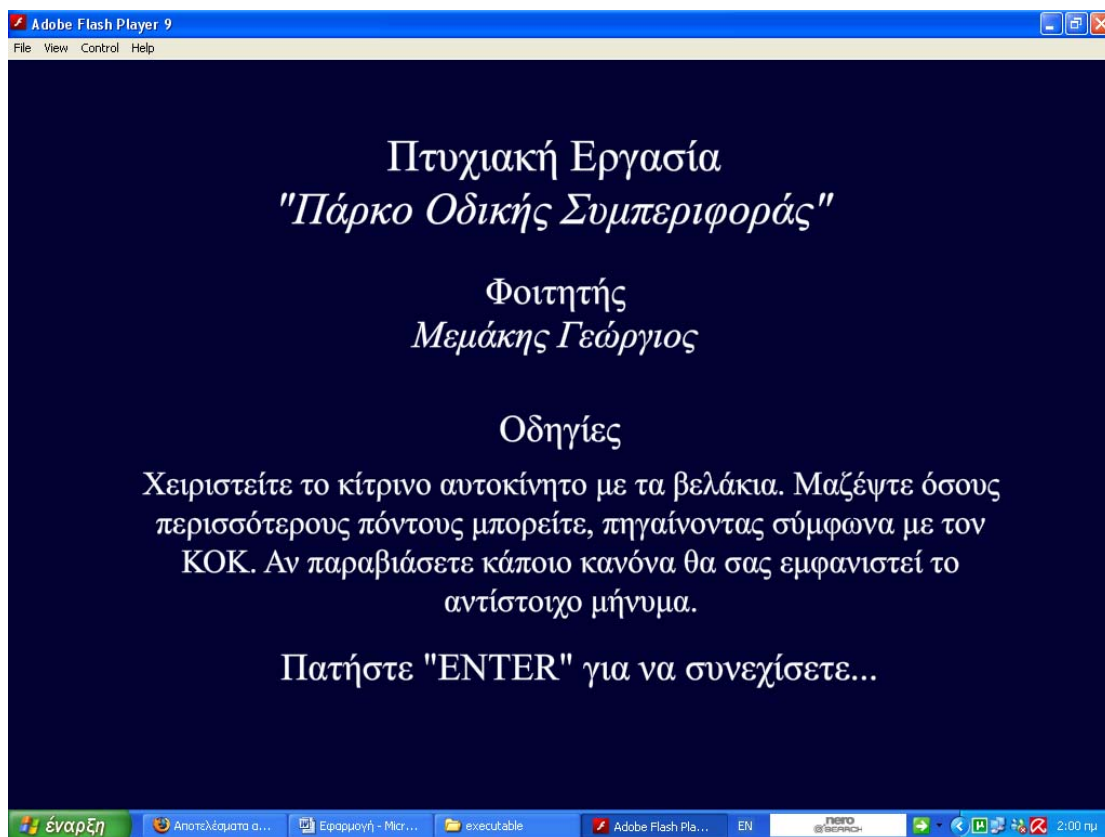
Στις 20 Αυγούστου του 2007, η Adobe ανήγγειλε στο blog της ότι με την αναπροσαρμογή 3 του Flash player, το βίντεο Flash θα υποστηρίξει επίσης το διεθνές πρότυπο mpeg-4. Συγκεκριμένα, ο Flash player θα έχει την υποστήριξη για το βίντεο που συμπιέζεται σε **H.264 (mpeg-4 μέρος 10)**, την ακουστική συμπιεσμένη χρησιμοποίηση **AAC (mpeg-4 μέρος 3)**, τα **MP4, M4V, M4A, 3GP** και εμπορευματοκιβωτίων πολυμέσων **MOV formats (mpeg-4 μέρος 14)**, **3GPP** χρονομετρημένη προδιαγραφή κειμένων (**mpeg-4 μέρος 17**) που είναι ένα τυποποιημένο **format** υποτίτλων και μια μερική υποστήριξη ανάλυσης για το 'list' που είναι τα **ID3** ισοδύναμα **iTunes** που χρησιμοποιεί για να αποθηκεύει τα μετά δεδομένα. Η Adobe επίσης ανήγγειλε ότι θα απομακρύνονται βαθμιαία από το ιδιόκτητο **format FLV** στο τυποποιημένο **MP4 format** εξ αιτίας των λειτουργικών ορίων με τη δομή FLV κατά τη ροή H.264. Η τελική καταβολή του Flash player που υποστηρίζει **mpeg-4** είχε διατεθεί το φθινόπωρο του 2007.

1.18 Κριτική

Οι κριτικές της Adobe Flash έχουν περιλάβει τα θέματα της δυνατότητας χρησιμοποίησής της, τα **Flash-laden** προβλήματα προκαλούν αναπηρίες στα θέματα ασφαλείας, περιορισμένα ζητήματα συμβατότητας πλατφορμών, απόδοσης και συμβατότητας σε ορισμένες πλατφόρμες, την ανικανότητα για τις μηχανές αναζήτησης να συνταχθούν τα στοιχεία που περιλαμβάνονται στα δυαδικά στοιχεία Flash, και τη χρήση της ως μέσο να περιοριστεί η πρόσβαση στο περιεχόμενο και η εφαρμογή της ψηφιακής διαχείρισης δικαιωμάτων. Υπάρχουν επίσης ανησυχίες μυστικότητας σχετικά με το τοπικό κοινό αντικείμενο (LSO), επίσης γνωστό ως **flash cookies**. Οι περισσότερες μηχανές αναζήτησης Ιστού έχουν τώρα τις δημοφιλείς επεκτάσεις που αποτρέπουν την άμεση αναπαραγωγή ήχου flash, και επιτρέπουν στο χρήστη είτε να αγνοήσει το flash ιστοσελίδας, είτε για να την παίξει με το κλικ σε αυτήν. Το Firefox έχει και **No Script** και **Flash block** ενώ μια χωριστή επέκταση για αποκαλούμενο όπερα **Flash block** είναι διαθέσιμη. Μια παρόμοια επέκταση για το Internet Explorer είναι **Foxie**, και περιέχει ένα αριθμό από **IE** χαρακτηριστικά γνωρίσματα, ένα από τα οποία ονομάζεται επίσης **Flash block**.

2. Εφαρμογή

Η εφαρμογή αυτή αναπτύχθηκε γεμάτη χρήσιμα πράγματα ώστε να βοηθήσει τον χρήστη να αποκτήσει σωστή Κυκλοφοριακή Αγωγή, δηλαδή όλα αυτά που χρειάζεται να ξέρουμε όλοι για τα σήματα και τις οδηγίες σωστής κυκλοφορίας στο δρόμο. Με την εφαρμογή αυτή, ο χρήστης μαθαίνει τα σήματα του Κώδικα Οδικής Κυκλοφορίας και την προφύλαξη από λάθη που οδηγούν σε τροχαία ατυχήματα. Η άγνοια, η απροσεξία, η βιασύνη είναι αφορμές που μπορούν να μας στερήσουν τη χαρά της ελευθερίας στις μετακινήσεις μας και την ομορφιά της ζωής. Η εφαρμογή προορίζεται κυρίως για παιδιά, για να μπορέσουν να μάθουν από μικρή ηλικία την σωστή κυκλοφοριακή αγωγή, αλλά και για ενήλικες που έχουν αποκτήσει άδεια οδήγησης προκειμένου να τους υπενθυμίσει κανόνες και οδηγίες σωστής κυκλοφορίας στο δρόμο. Αρχικά στην πρώτη οθόνη με την εκτέλεση της εφαρμογής μας παρουσιάζονται οι οδηγίες ο σκοπός και ο τρόπος εκτέλεσης της εφαρμογής.



Εικόνα 1: Παράθυρο εισαγωγής

Με την είσοδο μας στην εφαρμογή μας παρουσιάζεται μια μακέτα αποτελούμενη από αρκετά οικοδομικά τετράγωνα με κύριους και παράπλευρους δρόμους. Στην εφαρμογή μας έχουν εισαχθεί αυτοκίνητα ένα εκ των οποίων καλείται να χειριστή ο χρήστης ,φωτεινοί σηματοδότες , διαβάσεις πεζών, πεζοί, πεζοδρόμια , και διάφορες πινακίδες σήμανσης. Επίσης έχουμε εισάγει μουσικής ώστε η εφαρμογή μας να είναι ευχάριστη καθώς επίσης και ηχογραφήσεις για να μας υπενθυμίζουν τι πρέπει να κάνουμε. Τέλος έχουν προστεθεί διαγραμμίσεις στα οδικά δίκτυα, και ένας τροχονόμος για την ρύθμιση της κυκλοφορίας .



Εικόνα 2: Περιεχόμενα μακέτας

Το κίτρινο αυτοκίνητο που καλείτε να διαχειριστή ο χρήστης κινείται προς διάφορες κατευθύνσεις πατώντας τα αντίστοιχα βέλη στο κάτω δεξί τμήμα του πληκτρολογίου. Ταυτόχρονα στην εφαρμογή υπάρχουν και άλλα οχήματα τα οποία κινούνται αυτόνομα μέσα στα περιθώρια της μακέτας μας, και τα οποία επίσης τηρούν τον κώδικα οδικής κυκλοφορίας .Ο σκοπός για τον οποίον υπάρχουν στην μακέτα μας περισσότερα από ένα αυτοκίνητα είναι η εκμάθηση των κανόνων οδικής κυκλοφορίας και η καλλιέργεια της σωστής κυκλοφοριακής αγωγής στο χρήστη σε σχέση και με τους άλλους χρήστες των οδών , προσομοιάζοντας έτσι περισσότερο στην καθημερινή πραγματικότητα.

Αυτό φαίνεται και στην παρακάτω εικόνα:



Εικόνα 3: Τα οχήματα

2.1 Φωτεινοί σηματοδότες

Οι φωτεινοί σηματοδότες ρυθμίζουν τη κυκλοφορία στα σταυροδρόμια για τα οχήματα που υπάρχουν στην εφαρμογή μας. Η εισαγωγή τους κρίθηκε αναγκαία καθώς συναντιούνται συχνά στην οδική πραγματικότητα της εποχής μας. Βέβαια εδώ παρουσιάζονται σε απλοποιημένη μορφή κόκκινου και πράσινου χρώματος προκειμένου να είναι πιο εύχρηστοι και αντιληπτοί ιδιαίτερα στον νεαρό χρήστη.



Εικόνα 4: Φωτεινοί σηματοδότες

Έτσι παρατηρούμε ότι όταν κινούμε το αυτοκίνητο μας στον δρόμο που υπάρχει φανάρι τότε εμείς πρέπει να προσέξουμε το χρώμα του φωτεινού σηματοδότη ώστε να μην το παραβιάσουμε γιατί όταν το φανάρι είναι κόκκινο και πάμε να το περάσουμε τότε ο τροχονόμος μας λέει ότι «**σταμάτησε στο κόκκινο φανάρι**» και αυτόματα στην δεξιά οθόνη μας εμφανίζεται το παρακάτω σήμα:



Εικόνα 5: Παραβίαση φωτεινού σηματοδότη

2.2 Διαβάσεις πεζών

Επιλέχθηκαν για την εφαρμογή αυτή διαβάσεις πεζών τύπου ζέβρα προκειμένου να δείχνουν τα ασφαλή σημεία διάβασης των πεζών από το ένα πεζοδρόμιο στο άλλο. Ο σκοπός τους όσο αναφορά τον χρήστη της εφαρμογής είναι η οριοθέτηση και υπόδειξη ασφαλούς χώρου διακίνησης των πεζών και η διδασκαλία της προτεραιότητας και του σεβασμού απέναντι τους .



Εικόνα 6: Διαβάσεις πεζών.

Επίσης καθώς προχωράμε στο δρόμο μας και συναντήσουμε διάβαση πεζών , επί της οποίας κάποιος πεζός προσπαθήσει να περάσει την διάβαση αυτή τότε ο τροχονόμος θα του λέει το εξής: **προσοχή διάβαση πεζών** και αυτόματα εμείς δεν μπορούμε να προχωρήσουμε και αν το προσπαθήσουμε χάνουμε βαθμούς, και αυτό φαίνεται και στο δεξιό τμήμα της εφαρμογής μας με κόκκινο χρώμα. Αυτό φαίνεται στο παρακάτω σχήμα:



Εικόνα 7: Παραβίαση διάβασης πεζών

2.3 Πεζοδρόμια

Τα πεζοδρόμια εξασφαλίζουν την ασφαλή μετακίνηση των πεζών κατά μήκος των δρόμων. Αναπτύχθηκαν στην εφαρμογή μας καθότι σε όλους τους σύγχρονους δρόμους των πόλεων κατασκευάζονται πλατιά πεζοδρόμια για την μετακίνηση των πεζών.



Εικόνα 8: Πεζοδρόμια.

2.4 Πινακίδες σήμανσης

Οι πινακίδες σήμανσης ενημερώνουν προειδοποιούν και απαγορεύουν σε οδηγούς και πεζούς. Στην εφαρμογή μας επιλέχθηκαν ορισμένες αλλά βασικές πινακίδες σήμανσης όπως το STOP η απαγόρευση εισόδου για το όχημα, η απαγόρευση κίνησης προς μια ορισμένη κατεύθυνση, η ύπαρξη μονόδρομου κ.τ.λ. Επιλέχθηκαν καθώς αποτελούν μερικά από τα στοιχεία εκείνα που ρυθμίζουν και κάνουν ασφαλή την κυκλοφορία μας στους δρόμους.

Στην προκειμένη περίπτωση αν προσπαθήσουμε να εισέλθουμε σε ένα μονόδρομο ανάποδα τότε αμέσως ο τροχονόμος εκφωνεί ότι **απαγορεύεται να εισέλθεις σε αυτόν τον δρόμο**, και στην δεξιά πλευρά της εφαρμογής μας εμφανίζεται ένα απαγορευτικό, και αν προσπαθήσουμε να επιμείνουμε τότε οι βαθμοί μας αρχίζουν να μειώνονται αισθητά. Αυτό φαίνεται παρακάτω:



Εικόνα 9: Μονόδρομος



Εικόνα 10:Απαγόρευση εισόδου σε δρόμο.



Εικόνα 11:Υποχρεωτικό σταμάτημα (STOP)

2.5 Διαγραμμίσεις δρόμων

Παράλληλα επιλεχθήκαν δυο ειδών βασικές διαγραμμίσεις δρόμων προκειμένου ο χρήστης να διδαχθεί τα όρια κίνησης του οχήματος του στον δρόμο, την δυνατότητα προσπέρασης και τον σεβασμό στους άλλους χρήστες των οδών. Εμφανίζεται η διακεκομμένη διαγράμμιση οδού και η διπλή συνεχόμενη διαγραμμίσει.



Εικόνα 12: Διαγραμμίσεις οδών.

Τώρα όσο αναφορά την διπλή συνεχόμενη διαγράμμιση που υπάρχει στην εφαρμογή μας , εάν κατά την κίνηση του αυτοκινήτου μας πατήσουμε την διπλή αυτή διαγράμμιση τότε αμέσως ο τροχονόμος μας λέει ότι πηγαίνουμε ανάποδα στον μονόδρομο και συνάμα εμφανίζεται και δεξιά η ένδειξη λάθους καθώς και του ότι οι βαθμοί μας μειώνονται.

Αυτό φαίνεται παρακάτω:



Εικόνα 13: Παραβίαση διπλής διαγράμμισης

2.6 Τροχονόμος

Τέλος εισήχθη και ένας τροχονόμος για να ρυθμίζει την κυκλοφορία λόγος για τον οποίο εισήχθη είναι ότι απαντάται συχνά στο περιβάλλον των αστικών πόλεων και αποσκοπεί στην εκμάθηση της υπακοής και της πειθαρχίας στην κυκλοφοριακή αγωγή. Έτσι αν στην εφαρμογή μας προσπαθήσουμε να κάνουμε παραβιάσεις σε σχέση με τα σήματα ή τους φωτεινούς σηματοδότες ή στην διάβαση πεζών ή στην διπλή διαγράμμιση του δρόμου τότε αυτομάτως ο τροχονόμος μας λέει για το λάθος που κάναμε ή μας λέει τι να προσέξουμε.



Εικόνα 14:Υποδείξεις τροχονόμου.

Όπως προαναφέραμε ο χρήστης διαχειρίζεται το κίτρινο όχημα κατά την διάρκεια κυκλοφορίας του στους οδούς της μακέτας μας ,οφείλοντας να υπακούσει στους φωτεινούς σηματοδότες , τις πινακίδες κυκλοφορίας και στις υποδείξεις του τροχονόμου .Στην δεξιά πλευρά της οθόνης στην μακέτα μας εμφανίζεται ένα παράθυρο στο οποίο υποδεικνύονται οδηγίες για την σωστή κίνηση του οχήματος του χρήστη καθώς και επισημάνσεις εσφαλμένων κινήσεων. Επίσης στο παράθυρο αυτό επισημαίνεται με κόκκινο χρώμα η επικίνδυνη προσέγγιση σε πολύ κοντινή απόσταση με άλλο όχημα.

Τέλος προκειμένου η εφαρμογή μας να γίνει ιδιαίτερα φιλική, εκπαιδευτική και διασκευαστική από τους νέους σε ηλικία χρήστες της υπάρχει στο αναφερόμενο παράθυρο μια ένδειξη καταμέτρησης βαθμών που αυξάνονται ή μειώνονται ανάλογα με την ορθή ή εσφαλμένη κίνηση του οχήματος του χρήστη.

Μερικές ενδείξεις ανάλογα με την ορθή ή εσφαλμένη κίνηση του οχήματος του χρήστη φαίνονται παρακάτω:



1466
Σταμάτησε στο κόκκινο φανάρι



998
Πηγαίνετε ανάποδα στον μονόδρομο



1204
Προσοχή πεζός στην διάβαση!!!



686
Απαγορεύεται να εισέλθεις σε αυτό τον δρόμο



1162
Συνέχισε να πηγαίνεις σωστά στον δρόμο και κέρδισε περισσότερους βαθμούς

3. Ο Κώδικας

3.1 Κύρια αντικείμενα της σκηνής

```
var _MainStage = new MainStageSym();
this.addChild(_MainStage);
_MainStage.x = 110;
_MainStage.y = 110;

var _MainStageWindow = new MainStageWindowSym();
this.addChild(_MainStageWindow);

GlobalVarContainer.vars.sounds.stopSong();
var step:int = 5;
GlobalVarContainer.vars.sounds.playSong("../sounds//sound2.mp3");

var light1 = new LightSym();
_MainStage.addChild(light1);
light1.x = 272;
light1.y = 225;
light1.gotoAndPlay("fLightRed");

var light2 = new LightSym();
_MainStage.addChild(light2);
light2.x = 360;
light2.y = 225;
light2.gotoAndPlay("fLightRed");

var light3 = new LightSym();
_MainStage.addChild(light3);
light3.x = 380;
light3.y = 235;
light3.gotoAndPlay("fLightRed");

var light4 = new LightSym();
_MainStage.addChild(light4);
light4.x = 272;
light4.y = 319;
light4.gotoAndPlay("fLightRed");

var police = new PoliceSym();
_MainStage.addChild(police);
police.x = 320;
police.y = 275;

var myCar = new Car("Right", "Blue");
```

```
_MainStage.addChild(myCar);
myCar.x = 110;
myCar.y = 90;
var redCar = new Car("Down", "Red");
_MainStage.addChild(redCar);
redCar.x = 530;
redCar.y = 490;

var redCar2 = new Car("Right", "Red");
_MainStage.addChild(redCar2);
redCar2.x = 1050;
redCar2.y = 90;

var yellowCar = new Car("Right", "Yellow");
_MainStage.addChild(yellowCar);
yellowCar.x = 445;
yellowCar.y = 545;

var yellowCar2 = new Car("Right", "Yellow");
_MainStage.addChild(yellowCar2);
yellowCar2.x = 590;
yellowCar2.y = 90;

var messages:MessagesSymbol = new MessagesSymbol();
this.addChild(messages);
messages.x = 570;
messages.y = 298;
messages.gotoAndPlay("fStart");

var scoreInt:int;
scoreInt = 0;
var score = new ScoreSymbol();
this.addChild(score);
score.x = 572;
score.y = 200;
score.scoreText.text = scoreInt;

var people1 = new peopleSym();
_MainStage.addChild(people1);
people1.x = 735;
people1.y = 210;

var people2 = new peopleSym();
_MainStage.addChild(people2);
people2.x = 1300;
people2.y = 435;
```

Στον παραπάνω κώδικα περιγράφονται όλα τα αντικείμενα που υπάρχουν στην σκηνή μας (**Main Stage**). Όπως οι φωτεινοί σηματοδότες που ονομάζονται στην παρούσα φάση (**light1,light2,light3,light4**) και οι οποίοι βρίσκονται σε διάφορα σημεία της σκηνής μας. Επίσης έχουμε την δημιουργία του τροχονόμου που ονομάζεται **police**. Ακόμη το αυτοκινητάκι που χειρίζεται ο χρήστης που ονομάζεται **mycar**, καθώς και τα άλλα αυτοκινητάκια που υπάρχουν στην σκηνή μας και κινούνται ανεξάρτητα, και τα οποία ονομάζονται **redcar k' redcar2, yellowcar k' yellowcar2** αντίστοιχα. Ακόμη τα μηνύματα που εμφανίζονται στην δεξιά πλευρά της σκηνής μας (**messages**), και το **score**. Και τέλος έχουμε τους πεζούς οι οποίοι ονομάζονται **people**. Όλα τα παραπάνω αντικείμενα εισήλθαν στην σκηνή μας με την συνάρτηση **addchild()**.

3.2 Περιοδικές Συναρτήσεις

```
var timerPeriod2 = 2850;
var timerIsStarted2:Boolean;
var scanTimer2:Timer = null;
initTimer2(timerPeriod2);
startTimer2();

function initTimer2(iPeriod:int)
{
    scanTimer2 = new Timer(iPeriod);
    scanTimer2.addEventListener(TimerEvent.TIMER, startScan2_Handler, false, 0, true);
}

function startTimer2()
{
    timerIsStarted2 = true;
    scanTimer2.start();
}

function stopTimer2()
{
    timerIsStarted2 = false;
    scanTimer2.stop();
}

function startScan2_Handler($evt:TimerEvent):void
{
    changeLights();
}
```

```
//////////////////// Timer //////////////////////
var timerPeriod = 50;
var timerIsStarted:Boolean;
var scanTimer:Timer = null;
initTimer(timerPeriod);
startTimer();

function initTimer(iPeriod:int)
{
    scanTimer = new Timer(iPeriod);
    scanTimer.addEventListener(TimerEvent.TIMER, startScan_Handler, false, 0, true);
}

function startTimer()
{
    timerIsStarted = true;
    scanTimer.start();
}

function stopTimer()
{
    timerIsStarted = false;
    scanTimer.stop();
}

function startScan_Handler($evt:TimerEvent):void
{
    moveRedCar();
    moveRed2Car();
    moveYellowCar();
    moveYellow2Car();
    movePeople();
}
```

Στο παραπάνω κομμάτι κώδικα έχουμε δυο περιοδικές συναρτήσεις τον **Timer** , και τον **Timer2** , οι οποίοι καλούν περιοδικά αντίστοιχα τις συναρτήσεις **startScan_Handler** και **startScan2_Handler**. Όπου η **startScan_Handler** κινεί τα ανεξάρτητα αυτοκινητάκια και τους πεζούς. Ενώ η **startScan2_Handler** αλλάζει ανά τακτά χρονικά διαστήματα το χρώμα των φαναριών που υπάρχουν στην εφαρμογή μας .

3.3 Έλεγχος επιτρεπόμενης κίνησης

```
function isValidMovement(iDimX:int, iDimY:int, iCarIndex:int):Boolean
{
    if(iCarIndex !=1 && iCarIndex !=2)
    {
```



```
if(iDimX < 27 || iDimX > 1557 || iDimY < 27 || iDimY > 860)
    return false;

// first map
if(iDimX > 130 && iDimY > 110 && iDimX < 280 && iDimY < 240)
    return false;

if(iDimX > 355 && iDimY > 110 && iDimX < 505 && iDimY < 240)
    return false;

if(iDimX > 130 && iDimY > 310 && iDimX < 280 && iDimY < 460)
    return false;

if(iDimX > 355 && iDimY > 310 && iDimX < 505 && iDimY < 460)
    return false;

if(iDimX > 130 && iDimY > 570 && iDimX < 280 && iDimY < 735)
    return false;

if(iDimX > 355 && iDimY > 570 && iDimX < 505 && iDimY < 735)
    return false;

var secondMapOffset = 472;
// second map
if(iDimX > (130 + secondMapOffset) && iDimY > 110 && iDimX < (280 +
secondMapOffset) && iDimY < 240)
    return false;

if(iDimX > (355 + secondMapOffset) && iDimY > 110 && iDimX < (505 +
secondMapOffset) && iDimY < 240)
    return false;

if(iDimX > (130 + secondMapOffset) && iDimY > 310 && iDimX < (280 +
secondMapOffset) && iDimY < 460)
    return false;

if(iDimX > (355 + secondMapOffset) && iDimY > 310 && iDimX < (505 +
secondMapOffset) && iDimY < 460)
    return false;

if(iDimX > (130 + secondMapOffset) && iDimY > 570 && iDimX < (280 +
secondMapOffset) && iDimY < 735)
    return false;

if(iDimX > (355 + secondMapOffset) && iDimY > 570 && iDimX < (505 +
secondMapOffset) && iDimY < 735)
    return false;
```

```
        var thirdMapOffset = 2*472;
        // third map
        if(iDimX > (130 + thirdMapOffset) && iDimY > 110 && iDimX < (280 +
thirdMapOffset) && iDimY < 240)
            return false;

        if(iDimX > (355 + thirdMapOffset) && iDimY > 110 && iDimX < (505 +
thirdMapOffset) && iDimY < 240)
            return false;

        if(iDimX > (130 + thirdMapOffset) && iDimY > 310 && iDimX < (280 +
thirdMapOffset) && iDimY < 460)
            return false;

        if(iDimX > (355 + thirdMapOffset) && iDimY > 310 && iDimX < (505 +
thirdMapOffset) && iDimY < 460)
            return false;

        if(iDimX > (130 + thirdMapOffset) && iDimY > 570 && iDimX < (280 +
thirdMapOffset) && iDimY < 735)
            return false;

        if(iDimX > (355 + thirdMapOffset) && iDimY > 570 && iDimX < (505 +
thirdMapOffset) && iDimY < 735)
            return false;
    }

    if(iCarIndex !=0)
    {
        if(iDimX > myCar.x - myCar.width && iDimY > myCar.y - myCar.height &&
iDimX < myCar.x + myCar.width && iDimY < myCar.y + myCar.height)
            return false;
    }

    if(iCarIndex !=1)
    {
        if(iDimX > yellowCar.x - yellowCar.width && iDimY > yellowCar.y -
yellowCar.height &&
        iDimX < yellowCar.x + yellowCar.width && iDimY < yellowCar.y +
yellowCar.height)
        {
            if(iCarIndex ==0)
            {
                scoreInt--;
                score.scoreText.text = scoreInt;

                if(messages.currentLabel != "fPanelOtherCar")
                {
```

```
        messages.gotoAndPlay("fPanelOtherCar");

        GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\car.mp3");
    }
}
return false;
}
}

if(iCarIndex !=2)
{
    if(iDimX > redCar.x - redCar.width && iDimY > redCar.y - redCar.height &&
redCar.height)
        iDimX < redCar.x + redCar.width && iDimY < redCar.y +
    {
        if(iCarIndex ==0)
        {
            scoreInt--;
            score.scoreText.text = scoreInt;

            if(messages.currentLabel != "fPanelOtherCar")
            {
                messages.gotoAndPlay("fPanelOtherCar");

                GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\car.mp3");
            }
        }
        return false;
    }
}

if(iCarIndex !=3)
{
    if(iDimX > redCar2.x - redCar2.width && iDimY > redCar2.y - redCar2.height
&&
redCar2.height)
        iDimX < redCar2.x + redCar2.width && iDimY < redCar2.y +
    {
        if(iCarIndex ==0)
        {
            scoreInt--;
            score.scoreText.text = scoreInt;

            if(messages.currentLabel != "fPanelOtherCar")
            {
                messages.gotoAndPlay("fPanelOtherCar");

                GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\car.mp3");
            }
        }
    }
}
```

```
        }
    }
    return false;
}
}

if(iCarIndex !=4)
{
    if(iDimX > yellowCar2.x - yellowCar2.width && iDimY > yellowCar2.y -
yellowCar2.height &&
        iDimX < yellowCar2.x + yellowCar2.width && iDimY < yellowCar2.y +
yellowCar2.height)
    {
        if(iCarIndex ==0)
        {
            scoreInt--;
            score.scoreText.text = scoreInt;

            if(messages.currentLabel != "fPanelOtherCar")
            {
                messages.gotoAndPlay("fPanelOtherCar");

                GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\car.mp3");
            }
        }
        return false;
    }
}

if(iDimX >= 155 && iDimX <= 485 && iDimY >= 254 && iDimY <= 294)
{
    if(iCarIndex ==0)
    {
        if( myCar.getDirection() == "Right" && iDimY != 294 ||
myCar.getDirection() == "Left" && iDimY != 254)
        {
            scoreInt--;
            score.scoreText.text = scoreInt;

            if(messages.currentLabel != "fPanelOneway")
            {
                messages.gotoAndPlay("fPanelOneway");

                GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\oneway.mp3");
            }

            return false;
        }
    }
}
```

```
    }  
  }  
  
  if(iDimX >= 300 && iDimX <= 338 && iDimY >= 180 && iDimY <= 185 &&  
light1.currentLabel == "fLightRed")  
  {  
    if(iCarIndex ==0)  
    {  
      scoreInt--;  
      score.scoreText.text = scoreInt;  
  
      if(messages.currentLabel != "fPanelRedLight")  
      {  
        messages.gotoAndPlay("fPanelRedLight");  
  
GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\redlight.mp3");  
      }  
    }  
  
    return false;  
  }  
  
  if(iDimX >= 213 && iDimX <= 217 && iDimY >= 293 && iDimY <= 296 &&  
light4.currentLabel == "fLightRed")  
  {  
    if(iCarIndex ==0)  
    {  
      scoreInt--;  
      score.scoreText.text = scoreInt;  
  
      if(messages.currentLabel != "fPanelRedLight")  
      {  
        messages.gotoAndPlay("fPanelRedLight");  
  
GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\redlight.mp3");  
      }  
    }  
  
    return false;  
  }  
  
  if(iDimX >= 423 && iDimX <= 426 && iDimY >= 253 && iDimY <= 255 &&  
light3.currentLabel == "fLightRed")  
  {  
    if(iCarIndex ==0)  
    {  
      scoreInt--;  
      score.scoreText.text = scoreInt;
```

```
        if(messages.currentLabel != "fPanelRedLight")
        {
            messages.gotoAndPlay("fPanelRedLight");

            GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\redlight.mp3");
        }
    }
    return false;
}

if(iCarIndex == 0)
{
    //if(iDimX >= 295 && iDimX <= 345 && iDimY >= 90 && iDimY <= 470 &&
myCar.getDirection() == "Up")
    if(iDimX >= 295 && iDimX <= 345 && iDimY >= 90 && iDimY <= 735 &&
myCar.getDirection() == "Up")
    {
        scoreInt--;
        score.scoreText.text = scoreInt;

        if(messages.currentLabel != "fPanelDontEnter")
        {
            messages.gotoAndPlay("fPanelDontEnter");

            GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\dont.mp3");
        }

        return false;
    }

    //var secondMapOffset = 472;
    // second map
    if(iDimX >= (295 + secondMapOffset) && iDimX <= (345 + secondMapOffset)
&& iDimY >= 90 && iDimY <= 735 && myCar.getDirection() == "Up")
    {
        scoreInt--;
        score.scoreText.text = scoreInt;

        if(messages.currentLabel != "fPanelDontEnter")
        {
            messages.gotoAndPlay("fPanelDontEnter");

            GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\dont.mp3");
        }

        return false;
    }
}
```

```
//var thirdMapOffset = 2*472;
// third map
if(iDimX >= (295 + thirdMapOffset) && iDimX <= (345 + thirdMapOffset) &&
iDimY >= 90 && iDimY <= 735 && myCar.getDirection() == "Up")
{
    scoreInt--;
    score.scoreText.text = scoreInt;

    if(messages.currentLabel != "fPanelDontEnter")
    {
        messages.gotoAndPlay("fPanelDontEnter");

GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\dont.mp3");
    }

    return false;
}

if(iDimX > people1.x - people1.width && iDimY > people1.y - people1.height &&
iDimX < people1.x + people1.width && iDimY < people1.y + people1.height)
{
    if(iCarIndex ==0)
    {
        scoreInt--;
        score.scoreText.text = scoreInt;

        if(messages.currentLabel != "fPanelPeople")
        {
            messages.gotoAndPlay("fPanelPeople");

GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\people.mp3");
        }
    }
    return false;
}

if(iDimX > people2.x - people2.width && iDimY > people2.y - people2.height &&
iDimX < people2.x + people2.width && iDimY < people2.y + people2.height)
{
    if(iCarIndex ==0)
    {
        scoreInt--;
        score.scoreText.text = scoreInt;

        if(messages.currentLabel != "fPanelPeople")
        {
            messages.gotoAndPlay("fPanelPeople");
```

```
GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\people.mp3");
    }
    }
    return false;
}

if(messages.currentLabel != "fPanelOk" && iCarIndex == 0)
{
    messages.gotoAndPlay("fPanelOk");

GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\correct.mp3");
}

if(iCarIndex == 0)
{
    scoreInt++;
    score.scoreText.text = scoreInt;
}

return true;
}
```

Στο παρόν κομμάτι κώδικα έχουμε τον έλεγχο της κίνησης , που σημαίνει δηλαδή αν οι κινήσεις που κάνει ο χρήστης είναι ή δεν είναι επιτρεπτές τότε εμφανίζεται ένα μήνυμα στα δεξιά της σκηνής μας για τα λάθη που έκανε, καθώς επίσης και ποιος ήχος θα παιχτεί προκειμένου να υπενθυμίσει στον χρήστη την λάθος κίνηση. Τα μηνύματα εδώ εμφανίζονται με την εντολή **messages.gotoAndPlay("fPanelOk");** αν ο χρήστης πηγαίνει σωστά και με την εντολή **messages.gotoAndPlay("fPanelDontEnter");** για παράδειγμα ,αν απαγορεύεται να εισέλθει σε κάποιο δρόμο. Ακόμη οι ήχοι υπενθυμίσεις εισαγωντε με την εντολή **GlobalVarContainer.vars.MessagesSound.playSong(".\\sounds\\correct.mp3");** . Από την άλλη υπάρχει και μια μεταβλητή **CarIndex** η οποία μας βοηθά στο να μην πέφτουν τα ανεξάρτητα αυτοκινητάκια το ένα πάνω στο άλλο. Και τέλος υπάρχει μια μεταβλητή η **scoreInt** η οποία αυξάνει ή μειώνει το σκορ ανάλογα με τις κινήσεις που κάνει ο χρήστης , δηλαδή το αυξάνει αν ο χρήστης πηγαίνει σωστά και το μειώνει αν κάνει κινήσεις που δεν επιτρέπονται.

3.4 Κίνηση Αυτοκινήτου από το πληκτρολόγιο

```
stage.addEventListener(KeyboardEvent.KEY_DOWN, keyDownHandler);
function keyDownHandler(event:KeyboardEvent):void
{
    switch (event.keyCode)
    {
        case Keyboard.RIGHT:
            if( myCar.getDirection() == "Right")
            {
```



```
        if(isValidMovement(myCar.x + step + myCar.width/2,
myCar.y, 0) != false)
            {
                myCar.x += step;
                _MainStage.x -= step + 1;
            }
        }
        else
            myCar.setDirection("Right");
        break;
    case Keyboard.LEFT:
        if( myCar.getDirection() == "Left")
            {
                if(isValidMovement(myCar.x - step - myCar.width/2,
myCar.y, 0) != false)
                    {
                        myCar.x -= step;
                        _MainStage.x += step+1;
                    }
            }
        }
        else
            myCar.setDirection("Left");
        break;
    case Keyboard.UP:
        if( myCar.getDirection() == "Up")
            {
                if(isValidMovement(myCar.x, myCar.y - step -
myCar.height/2, 0) != false)
                    {
                        myCar.y -= step;
                        _MainStage.y += step+2;
                    }
            }
        }
        else
            myCar.setDirection("Up");
        break;
    case Keyboard.DOWN:
        if( myCar.getDirection() == "Down")
            {
                if(isValidMovement(myCar.x, myCar.y + step +
myCar.height/2, 0) != false)
                    {
                        myCar.y += step;
                        _MainStage.y -= step+2;
                    }
            }
        }
        else
            myCar.setDirection("Down");
```

```
        break;
    case 83: //s
        _MainStage.y += 50;
        break;
    case 68: //d
        _MainStage.x += 50;
        break;
    case 65: //a
        _MainStage.x -= 50;
        break;
    case 87: //w
        _MainStage.y -= 50;
        break;
    default:
        break;
}
}
```

Στο παραπάνω κομμάτι κώδικα ελέγχονται οι κινήσεις του αυτοκινήτου που κινεί ο χρήστης. Δηλαδή ποιες κινήσεις μπορεί να κάνει και με ποια πλήκτρα. Όπως καταλαβαίνουμε εδώ τα πλήκτρα που μας βοηθάνε να κινούμε το αυτοκινητάκι μας είναι τα βελάκια που υπάρχουν στο πληκτρολόγιο μας. Ακόμη μπορεί να ελέγχει και την ορθότητα των κινήσεων του αμαξιού, δηλαδή αν όντως επιτρέπεται να κάνει την κίνηση αυτή με βάση τα πλήκτρα που αναφέραμε παραπάνω. Η συνάρτηση με την οποία μπορώ να προσδώσω στο αυτοκινητάκι μου την κίνηση δεξιά αριστερά, πάνω ή κάτω εκτελείτε με την εντολή **keyDownHandle** καθώς και τον αν η κίνηση μας είναι ορθή γίνεται με την συνάρτηση **isValidMovement**. Επίσης έχουμε προσθέσει και τα πλήκτρα s,d,a,w έτσι ώστε να μετακινούμε την μακέτα μας όπου εμείς θέλουμε ώστε να βλέπουμε που θέλουμε να μετακινηθούμε.

3.5 Αντικείμενο Car

```
package
{
    import flash.display.*;

    public class Car extends Sprite
    {
        var _CarSprite:CarSym;
        var _CarCurrDirection:String;
        var _CarColor:String;

        public function Car(iDirection:String, iColor:String)
        {
            _CarSprite = new CarSym();
            this.addChild(_CarSprite);
        }
    }
}
```

```
        _CarCurrDirection = iDirection;
        _CarColor = iColor;

        _CarSprite.gotoAndPlay("fCar" + _CarColor + _CarCurrDirection);
    }

    public function setDirection(iDirection:String)
    {
        _CarCurrDirection = iDirection;
        _CarSprite.gotoAndPlay("fCar" + _CarColor + _CarCurrDirection);
    }

    public function getDirection():String
    { return _CarCurrDirection; }
}
}
```

Στο σημείο αυτό περιέχονται πληροφορίες σχετικά με το αυτοκινητάκι που χρησιμοποιεί ο χρήστης .Δηλαδή το χρώμα που του προσδίδουμε καθώς και την κατεύθυνση του, και γενικά την εικόνα που έχει στη εφαρμογή μας.

3.6 Διαχείριση ήχου

```
package
{
    import flash.media.Sound;
    import flash.media.SoundChannel;
    import flash.media.SoundMixer;
    import flash.media.SoundTransform;
    import flash.utils.*;
    import flash.net.URLRequest;
    import flash.events.*;
    import flash.filters.*;
    import flash.errors.IOError;

    public class AudioManager
    {
        private var _sound: Sound;
        private var _sc: SoundChannel;

        private var _index:int;
        private var soundsArray:Array;

        private var _soundsQueue:Array;

        public function AudioManager()
        {
```

```
        soundsArray = null;
    _sc = new SoundChannel();
    _sound = new Sound();
    _sound.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
    _sc.addEventListener(Event.SOUND_COMPLETE, onComplete);
    }

    public function onComplete(e:Event):void
    {
        _sc.stop();
        _index = 0;
    }

    public function multipleSoundsPlayer(soundsQueue:Array)
    {
        _sc.stop();
        _soundsQueue = soundsQueue;
        playNext();
    }

    private function playNext(event:Event = null)
    {
        if (_soundsQueue != null && _soundsQueue.length > 0)
        {
            _sound = new Sound();
            _sound.load(new URLRequest(_soundsQueue.shift()));
            _sc = _sound.play();
            _sc.addEventListener(Event.SOUND_COMPLETE, playNext,
false,0,true);
        }
    }

    private function ioErrorHandler(event: IOErrorEvent): void
    {
    }

    public function playSong(iSoundURL:String)
    {
        soundsArray = null;
        _sc.stop();
        _sound = new Sound();
        _sound.load(new URLRequest(iSoundURL));
        _sc = _sound.play();
        _sc.addEventListener(Event.SOUND_COMPLETE, onComplete);
        _sound.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
    }

    public function playSongs(iSoundURLs:String)
    {
```

```
        soundsArray = iSoundURLs.split(',');

        _sc.stop();
        if(soundsArray.length == 1)
        {
            this.playSong(iSoundURLs);
        }
        else
        {
            _sound = new Sound();
            _sound.load(new URLRequest(soundsArray[_index]));
            _sc = _sound.play();
            _sc.addEventListener(Event.SOUND_COMPLETE,
onCompleteFirst);
            _sound.addEventListener(IOErrorEvent.IO_ERROR,
ioErrorHandler);
        }
    }

    public function onCompleteFirst(e:Event):void
    {
        _sc.stop();
        ++_index;
        _sound = new Sound();
        _sound.load(new URLRequest(soundsArray[_index]));
        _sc = _sound.play();
        _sc.addEventListener(Event.SOUND_COMPLETE, onComplete);
        _sound.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
    }

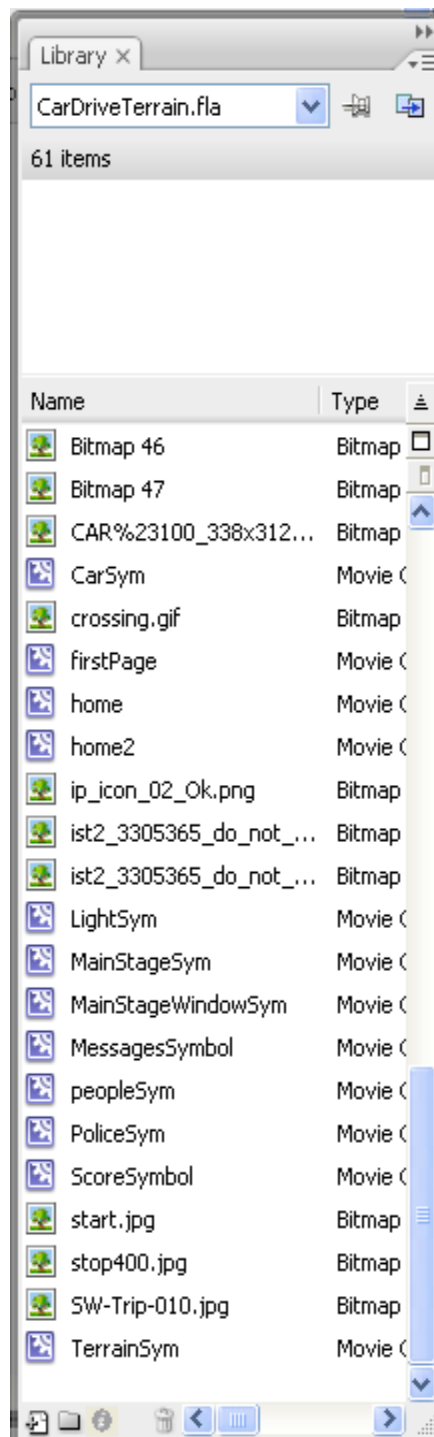
    public function stopSong()
    { _sc.stop(); }

    public function pauseSong()
    { _sc.stop(); }

    public function resumeSong()
    { _sc = _sound.play(); }
}
```

Στο σημείο αυτό ο παραπάνω κώδικας ελέγχει τον ήχο της εφαρμογής μας γενικά. Πότε δηλαδή θα παίζει ο ήχος πότε θα σταματάει . Τα ονόματα των συναρτήσεων είναι χαρακτηριστικά. Απλά έχουμε **sound** και **channels**. Όπου **sound** είναι οι ήχοι που παίζουν στην εφαρμογή μας ,και τα **channels** είναι αυτά μέσω των οποίων παρακολουθείται ο ήχος που παίζεται κάθε φορά.

3.7 Βιβλιοθήκη Flash



Τα βασικά αντικείμενα και μερικές από τις εικόνες που χρησιμοποιήθηκαν σε αυτή την εργασία. Εικόνα από την βιβλιοθήκη της εφαρμογής.

4. Επίλογος

Η εφαρμογή αυτή δημιουργήθηκε με το πρόγραμμα **Adobe Flash CS3 Professional** και με γλώσσα προγραμματισμού **Action Script 3.0**.

Η **Action Script 3.0** στάθηκε πολύτιμο εργαλείο κα βοήθημα στη δημιουργία της συμπεριφοράς των κινούμενων αντικειμένων της μακέτας. Εισήχθησαν συναρτήσεις για την κίνηση των αντικειμένων της σκηνής (ανεξάρτητα αυτοκίνητα, φωτεινοί σηματοδότες, πεζοί κτλ) και του οχήματος του χρήστη καθώς και για τα χρονικά διαστήματα αλλαγής χρωμάτων στους φωτεινούς σηματοδότες. Επιλέχθηκαν οι κατάλληλες εντολές για τα αντίστοιχα μηνύματα που εμφανίζονται στην δεξιά πλευρά της οθόνης, ανάλογα με την κάθε φορά κίνηση του οχήματος του χρήστη.

Η χρήση των σωστών μεταβλητών επέτρεψαν την καταμέτρηση και αυξομείωση του σκορ του χρήστη σύμφωνα πάντα με τη κίνηση του οχήματος του. Οι χρησιμοποιούμενοι κώδικες ελέγχουν την ακρόαση ή την παύση του ήχου της εφαρμογής μας ενώ αποτελούν μέσο για την παρακολούθηση του ήχου που παίζεται κάθε φορά.

Οι τύποι των αρχείων που χρησιμοποιήθηκαν είναι κυρίως **.SWF** και **.fla**

Από την πρώτη οθόνη της εφαρμογής μας παρουσιάζονται οι οδηγίες ο σκοπός και ο τρόπος εκτέλεσης της εφαρμογής. Έγινε προσπάθεια να σχεδιαστεί όσο το δυνατόν ρεαλιστικότερα με την πραγματικότητα του σήμερα και να αποδοθεί όσο το δυνατόν πιστότερα και απλοποιημένα πάντα το σύγχρονο οδικό περιβάλλον των πόλεων.

Έτσι στην εφαρμογή μας παρουσιάζεται μια μακέτα αποτελούμενη από αρκετά οικοδομικά τετράγωνα με κύριους και παράπλευρους δρόμους. Έχουν εισαχθεί αυτοκίνητα, φωτεινοί σηματοδότες, διαβάσεις πεζών, πεζοί, πεζοδρόμια και διάφορες πινακίδες σήμανσης ενώ έχουν προστεθεί διαγραμμίσεις στα οδικά δίκτυα και ένας τροχονόμος για την ρύθμιση της κυκλοφορίας.

Η χώρα μας κατέχει μια από τις θλιβερές πρωτιές την ευρωπαϊκή ένωση στον αριθμό των ατυχημάτων και των θανάτων και τραυματισμών σε τροχαία δυστυχήματα. Η άγνοια, η απροσεξία, η βιασύνη είναι αφορμές που μπορούν να μας στερήσουν τη χαρά της ελευθερίας στις μετακινήσεις μας και την ομορφιά της ζωής.

Η εφαρμογή απευθύνεται κυρίως σε μικρά παιδιά και συνεισφέρει στην ανάπτυξη δομών της προσωπικότητας και συμπεριφοράς στον ευαίσθητο για την χώρα μας τομέα αυτό.

Ταυτόχρονα σε πιο εξελιγμένη μορφή θα μπορούσε κάλλιστα να αποτελέσει βασικό εργαλείο υπενθύμισης κανόνων και οδηγιών σωστής κυκλοφορίας στο δρόμο και καλλιέργειας της ορθής κυκλοφορικής αγωγής.

Γενικά είναι ένα δώρο πρώτα απ' όλα για τα παιδιά μας, το μέλλον της κοινωνίας μας, αλλά και ένα δώρο για όλους.

5. Παραρτήματα

- Flash CS3 For Dummies May 2007_(βιβλίο)
- Macromedia www.macromedia.com/support/flash/
- Moock.org www.moock.org
- OS Flash <http://osflash.org/>
- Flash Kit <http://www.flashkit.com/>
- ActionScript.org www.actionscript.org
- ActionScript.com www.actionscript.com
- FlashComponents.net www.flashcomponents.net