

**ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ**

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ**

**ΚΑΤΕΥΘΥΝΣΗ: «ΠΡΟΗΓΜΕΝΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ &  
ΕΦΑΡΜΟΓΕΣ»**

**ΘΕΜΑ ΔΙΠΛΩΜΑΤΙΚΗΣ**

**Σχεδιασμός και ανάπτυξη συστήματος για video on-demand από θεατρικές  
παραστάσεις σε έξυπνες τηλεοράσεις (Smart TVs)**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Γεώργιος Δημητρακόπουλος, Λέκτορας**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΦΟΙΤΗΤΡΙΑ: Σοφία Αγραφιώτη, Α.Μ:12201**

**ΤΑΥΡΟΣ 2014**



## ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια υπάρχει ραγδαία ανάπτυξη του διαδικτύου και η καθημερινότητα μας κατακλύζεται από εφαρμογές του. Η έννοια του “smart” εμφανίζεται όλο και περισσότερο μετατρέποντας απλές συσκευές σε πιο σύνθετες, με περισσότερες λειτουργίες και δυνατότητες, παραδείγματος χάριν συσκευές, όπως οι τηλεοράσεις αποκτούν λειτουργικό αντίστοιχο των υπολογιστών και σύνδεση στο internet.

Η **έξυπνη τηλεόραση** είναι από τις τελευταίες τεχνολογικές εξελίξεις στο χώρο του Home Entertainment και περιγράφει μια τάση ενσωμάτωσης του Internet και των Web 2.0 χαρακτηριστικών, δηλαδή χρήστες χωρίς εξειδικευμένες γνώσεις αποκτούν ενεργό ρόλο και μπορούν να επεμβαίνουν στο περιεχόμενο των ιστοσελίδων, στις τηλεοράσεις και σε αποκωδικοποιητές (set-top boxes). Οι “έξυπνες” τηλεοράσεις προσφέρουν προηγμένη υπολογιστική ικανότητα και δυνατότητα σύνδεσης στο διαδίκτυο σε σχέση με αυτή της κλασικής τηλεόρασης, επίσης έχουν τη δυνατότητα να τρέχουν εφαρμογές (application).

Στην παρούσα εργασία μελετώνται πρωτόκολλα για Video on demand, πρωτόκολλα που χρησιμοποιούνται σε Smart TVs και τέλος αναπτύσσεται μια εφαρμογή (application) για Smart TV(έξυπνες τηλεοράσεις).

Με την εφαρμογή αυτή οι χρήστες θα έχουν τη δυνατότητα να παρακολουθούν μαγνητοσκοπημένες θεατρικές παραστάσεις από την τηλεόραση τους.

Αναλυτικότερα, η εργασία χωρίζεται σε 3 κεφάλαια:

1. Εισαγωγή
2. Θεωρητικό υπόβαθρο - έννοιες
3. Περιγραφή υλοποίησης.

Στο πρώτο κεφάλαιο πραγματοποιείται μια σύντομη βιβλιογραφική ανασκόπηση μέσω κάποιων paper και παρουσιάζονται ενδεικτικές έρευνες. Στο δεύτερο κεφάλαιο μελετούνται οι βασικές έννοιες που απαιτούνται για την κατανόηση της υλοποίησης. Στο τρίτο και τελευταίο κεφάλαιο γίνεται η παρουσίαση της εφαρμογής και των εργαλείων που χρησιμοποιήθηκαν.

## ABSTRACT

In the recent years there is a rapid development of the internet and our daily lives are consumed by our applications. The concept of "smart" is ever increasingly used, transforming simple devices into more complex ones with more functions and potentials, for example devices such as televisions are being developed with operating systems equivalent of computers and internet connection.

The **smart television** is one of the latest technological development in home entertainment and is indicative of a tendency to integrate the internet and web 2.0 characteristics, allowing users without specialized knowledge to gain an active role and are able to manipulate the content of the websites, on their television sets and as well as setup boxes. Smart televisions offer advanced computing power and connectivity to the world wide web in comparison to the classic analog television as well as being capable of running applications.

This paper is concerned with the study of video on demand protocols, protocols in smart televisions and finally an application is developed for smart televisions.

This application will allow users the potential to view video recorded theatrical shows from their television.

More specifically, this paper is sectioned into three parts.

1. Introduction
2. Theoretical background- concepts
3. Implementation description

The first chapter consists of a brief literature review and indicative surveys are presented. In the second chapter basic concepts required for the understanding of the development. In the third and final chapter the application is presented alongside the tools used.

## ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Multimedia(Πολυμέσα), Samsung smart TV,VoD, Πρωτόκολλα RTP/RTSP, Video playback

## KEYWORDS

Multimedia, Samsung smart TV,VoD, Protocols RTP/RTSP, Video playback

## ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία με θέμα: «Σχεδιασμός και ανάπτυξη συστήματος για video on-demand από θεατρικές παραστάσεις σε έξυπνες τηλεοράσεις (Smart TVs)» εκπονήθηκε στα πλαίσια της διπλωματικής μου εργασίας στο τμήμα «πληροφορικής και τηλεματικής» υπό την επίβλεψη του καθηγητή κ. Γεώργιου Δημητρακόπουλου.

Με την ολοκλήρωση των μεταπτυχιακών μου σπουδών στο τμήμα πληροφορικής και τηλεματικής στο Χαροκόπειο πανεπιστήμιο έχω αποκομίσει σημαντικές γνώσεις στο χώρο των σύγχρονων τηλεπικοινωνιακών συστημάτων και δικτύων.

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον Λέκτορα Καθηγητή και επιβλέποντα της πτυχιακής μου εργασίας κ. Γεώργιο Δημητρακόπουλο για την πολύτιμη βοήθεια και υποστήριξη του τόσο στο θεωρητικό όσο και στο τεχνικό επίπεδο.

Τέλος, θα ήθελα να ευχαριστήσω όλο το εκπαιδευτικό και διοικητικό προσωπικό του Πανεπιστημίου για την προσφορά και βοήθεια τους σε όλη τη διάρκεια των Μεταπτυχιακών σπουδών μου.

## ΠΕΡΙΕΧΟΜΕΝΑ

|                                                                                                                          |        |
|--------------------------------------------------------------------------------------------------------------------------|--------|
| ΠΕΡΙΛΗΨΗ .....                                                                                                           | - 3 -  |
| ABSTRACT .....                                                                                                           | - 4 -  |
| ΠΡΟΛΟΓΟΣ.....                                                                                                            | - 5 -  |
| ΠΕΡΙΕΧΟΜΕΝΑ .....                                                                                                        | - 6 -  |
| ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ .....                                                                                                   | - 7 -  |
| ΚΕΦΑΛΑΙΟ 1 <sup>ο</sup> ΕΙΣΑΓΩΓΗ .....                                                                                   | - 10 - |
| 1.1 Ερευνητικό πεδίο .....                                                                                               | - 10 - |
| 1.2 Επισκόπηση βιβλιογραφίας .....                                                                                       | - 11 - |
| 1 <sup>ο</sup> Television as an App:Taming the Device Fragmentation Problem .....                                        | - 11 - |
| 2 <sup>ο</sup> HTML5 and Companion Web Technologies as a Universal Platform for Interactive Internet TV Advertising..... | - 12 - |
| 3 <sup>ο</sup> Roles of Smart TV in IoT-environments: a Survey.....                                                      | - 13 - |
| 4 <sup>ο</sup> Smart TV Hacking: Crash Testing Your Home Entertainment .....                                             | - 15 - |
| 1.3 Έρευνες Nielsen - Eurostat .....                                                                                     | - 17 - |
| 1.3.1 Nielsen: Χρήση VoD.....                                                                                            | - 17 - |
| 1.3.2 Eurostat: «στατιστικά στον πολιτιστικό τομέα» EE-27 .....                                                          | - 18 - |
| 1.3.3 Αθηνόραμα.....                                                                                                     | - 20 - |
| ΚΕΦΑΛΑΙΟ 2 <sup>ο</sup> ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ - ΕΝΝΟΙΕΣ .....                                                               | - 21 - |
| 2.1 Ορισμοί.....                                                                                                         | - 21 - |
| Τύποι Streaming .....                                                                                                    | - 22 - |
| Πρωτόκολλα Streaming.....                                                                                                | - 22 - |
| Εφαρμογές τεχνικής Adaptive Bitrate streaming.....                                                                       | - 23 - |
| Τεχνολογίες που υποστηρίζει μια smart TV .....                                                                           | - 23 - |
| Κατηγορίες τηλεοράσεων .....                                                                                             | - 24 - |
| Πρωτόκολλα Smart Tv.....                                                                                                 | - 25 - |
| Οργανισμοί.....                                                                                                          | - 25 - |
| Νομικά.....                                                                                                              | - 25 - |
| 2.2 Κατασκευαστές .....                                                                                                  | - 26 - |
| ΚΕΦΑΛΑΙΟ 3 <sup>ο</sup> ΠΕΡΙΓΡΑΦΗ ΥΛΟΠΟΙΗΣΗΣ.....                                                                        | - 29 - |
| 3.1 Περιγραφή των εργαλείων.....                                                                                         | - 29 - |

|                                                    |      |
|----------------------------------------------------|------|
| 3.2 Δομή εφαρμογής - Περιγραφή.....                | 30 - |
| 3.2.1Διάρθρωση .....                               | 34 - |
| 3.2.2 Μέρη εφαρμογής .....                         | 35 - |
| 3.2.3 Περιγραφή κώδικα .....                       | 36 - |
| config.xml .....                                   | 36 - |
| index.html.....                                    | 37 - |
| Αρχικοποίηση εφαρμογής - main.js .....             | 40 - |
| Επικοινωνία με server .....                        | 48 - |
| Setter Getters methods .....                       | 50 - |
| XML/videoList.xml .....                            | 51 - |
| Λειτουργίες Video Playback Buffer .....            | 52 - |
| Διαχείριση - Εμφάνιση Λειτουργιών ήχου .....       | 59 - |
| Εμφάνιση όλων των αντικειμένων στην εφαρμογή ..... | 63 - |
| Common/API/ TVKeyValue.js .....                    | 71 - |
| Φάκελοι .....                                      | 73 - |
| ΣΥΜΠΕΡΑΣΜΑΤΑ .....                                 | 74 - |
| ΓΛΩΣΣΑΡΙ .....                                     | 75 - |
| ΒΙΒΛΙΟΓΡΑΦΙΑ .....                                 | 76 - |
| ΠΑΡΑΡΤΗΜΑ –ΚΩΔΙΚΑΣ .....                           | 77 - |

## ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

|                                                                   |      |
|-------------------------------------------------------------------|------|
| ΕΙΚΟΝΑ 1 ACTIVE VIDEO NETWORKS « <b>CLOUD TV</b> » APPROACH ..... | 11 - |
| ΕΙΚΟΝΑ 2 THE LANDING PAGE / BRANDED MICROSITE IN HTML5 .....      | 13 - |
| ΕΙΚΟΝΑ 3 .....                                                    | 14 - |
| ΕΙΚΟΝΑ 4 .....                                                    | 16 - |
| ΕΙΚΟΝΑ 5 “DIAGRAM BLACK-BOX TESTING”*** .....                     | 16 - |
| ΕΙΚΟΝΑ 6 .....                                                    | 17 - |
| ΕΙΚΟΝΑ 7 , ΠΗΓΗ: NIELSEN, 2013.....                               | 17 - |
| ΕΙΚΟΝΑ 8 .....                                                    | 17 - |
| ΕΙΚΟΝΑ 9 .....                                                    | 18 - |
| ΕΙΚΟΝΑ 10 .....                                                   | 18 - |
| ΕΙΚΟΝΑ 11 .....                                                   | 18 - |
| ΕΙΚΟΝΑ 12, ΠΗΓΗ: ΕΥΡΩΒΑΡΟΜΕΤΡΟ 67.1, 2007 .....                   | 19 - |

|                                                       |        |
|-------------------------------------------------------|--------|
| ΕΙΚΟΝΑ 13 .....                                       | - 20 - |
| ΕΙΚΟΝΑ 14 .....                                       | - 27 - |
| ΕΙΚΟΝΑ 15 .....                                       | - 27 - |
| ΕΙΚΟΝΑ 16 .....                                       | - 29 - |
| ΕΙΚΟΝΑ 17. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΜΙΑ ΕΞΥΠΝΗΣ ΤΗΛΕΟΡΑΣΗΣ ..... | - 30 - |
| ΕΙΚΟΝΑ 18. ΔΟΜΗ ΕΦΑΡΜΟΓΗΣ .....                       | - 30 - |
| ΕΙΚΟΝΑ 19 .....                                       | - 30 - |
| ΕΙΚΟΝΑ 20 .....                                       | - 31 - |
| ΕΙΚΟΝΑ 21 .....                                       | - 31 - |
| ΕΙΚΟΝΑ 22 .....                                       | - 32 - |
| ΕΙΚΟΝΑ 23 .....                                       | - 33 - |
| ΕΙΚΟΝΑ 24 .....                                       | - 33 - |
| ΕΙΚΟΝΑ 25 .....                                       | - 34 - |
| ΕΙΚΟΝΑ 26 .....                                       | - 35 - |
| ΕΙΚΟΝΑ 27 .....                                       | - 36 - |
| ΕΙΚΟΝΑ 28 .....                                       | - 37 - |
| ΕΙΚΟΝΑ 29 .....                                       | - 38 - |
| ΕΙΚΟΝΑ 30 .....                                       | - 38 - |
| ΕΙΚΟΝΑ 31 .....                                       | - 38 - |
| ΕΙΚΟΝΑ 32 .....                                       | - 39 - |
| ΕΙΚΟΝΑ 33 .....                                       | - 40 - |
| ΕΙΚΟΝΑ 34 .....                                       | - 41 - |
| ΕΙΚΟΝΑ 35 .....                                       | - 41 - |
| ΕΙΚΟΝΑ 36 .....                                       | - 42 - |
| ΕΙΚΟΝΑ 37 .....                                       | - 42 - |
| ΕΙΚΟΝΑ 38 .....                                       | - 44 - |
| ΕΙΚΟΝΑ 39 .....                                       | - 45 - |
| ΕΙΚΟΝΑ 40 .....                                       | - 45 - |
| ΕΙΚΟΝΑ 41 .....                                       | - 46 - |
| ΕΙΚΟΝΑ 42 .....                                       | - 47 - |
| ΕΙΚΟΝΑ 43 .....                                       | - 47 - |
| ΕΙΚΟΝΑ 44 .....                                       | - 47 - |
| ΕΙΚΟΝΑ 45 .....                                       | - 48 - |
| ΕΙΚΟΝΑ 46 .....                                       | - 49 - |
| ΕΙΚΟΝΑ 47 .....                                       | - 50 - |
| ΕΙΚΟΝΑ 48 .....                                       | - 51 - |
| ΕΙΚΟΝΑ 49 .....                                       | - 52 - |

|                                                                                    |        |
|------------------------------------------------------------------------------------|--------|
| EIKONA 50 .....                                                                    | - 52 - |
| EIKONA 51 .....                                                                    | - 53 - |
| EIKONA 52 .....                                                                    | - 54 - |
| EIKONA 53 .....                                                                    | - 54 - |
| EIKONA 54 .....                                                                    | - 54 - |
| EIKONA 55 ΜΕΤΑΒΑΣΗ ΑΠΟ ΤΗΝ ΚΑΤΑΣΤΑΣΗ PLAY → PAUSE → ΣΤΗΝ PLAY – RESUME → STOP .... | - 56 - |
| EIKONA 56 .....                                                                    | - 57 - |
| EIKONA 57 .....                                                                    | - 58 - |
| EIKONA 58 .....                                                                    | - 59 - |
| EIKONA 59 .....                                                                    | - 59 - |
| EIKONA 60 .....                                                                    | - 60 - |
| EIKONA 61 .....                                                                    | - 61 - |
| EIKONA 62 .....                                                                    | - 61 - |
| EIKONA 63 .....                                                                    | - 62 - |
| EIKONA 64 .....                                                                    | - 62 - |
| EIKONA 65 .....                                                                    | - 63 - |
| EIKONA 66 .....                                                                    | - 63 - |
| EIKONA 67 .....                                                                    | - 63 - |
| EIKONA 68 .....                                                                    | - 64 - |
| EIKONA 69 .....                                                                    | - 65 - |
| EIKONA 70 .....                                                                    | - 66 - |
| EIKONA 71 .....                                                                    | - 67 - |
| EIKONA 72 .....                                                                    | - 67 - |
| EIKONA 73 .....                                                                    | - 67 - |
| EIKONA 74 .....                                                                    | - 68 - |
| EIKONA 75 .....                                                                    | - 68 - |
| EIKONA 76 .....                                                                    | - 68 - |
| EIKONA 77 .....                                                                    | - 70 - |
| EIKONA 78 .....                                                                    | - 71 - |
| EIKONA 79 .....                                                                    | - 71 - |
| EIKONA 80 .....                                                                    | - 72 - |
| EIKONA 81 .....                                                                    | - 73 - |
| EIKONA 82 .....                                                                    | - 73 - |

## ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> ΕΙΣΑΓΩΓΗ

### 1.1 Ερευνητικό πεδίο

Η δημιουργία της έξυπνης τηλεόρασης αποτελεί μια καινοτόμα λύση με ποικίλες εφαρμογές στο οικιακό περιβάλλον. Τις πρώτες χρήσεις της συναντάμε στο χώρο του Home Entrainment και της διαφήμισης. Από τις πιο ενδιαφέρουσες υπηρεσίες, όσον αφορά τη διασκέδαση, αποτελεί η υπηρεσία streaming περιεχομένου είτε σε πραγματικό χρόνο(real time) είτε κατόπιν ζήτησης περιεχομένου(on demand).

Βασικό πλεονέκτημα των υπηρεσιών streaming αποτελεί το γεγονός ότι η αναπαραγωγή του video μπορεί να πραγματοποιηθεί αμέσως, αντί να αποθηκεύεται στο σκληρό δίσκο. Επιπλέον ο χρήστης μπορεί να κάνει χρήση των λειτουργιών: play, pause και resume , όποτε το επιθυμεί.

Με τις έξυπνες τηλεοράσεις υπάρχει η δυνατότητα πρόσβασης στο διαδίκτυο με αποτέλεσμα να ανοίγονται νέοι δρόμοι και στο χώρο της διαφήμισης, αφού δίνεται η δυνατότητα στον θεατή να αλληλεπιδρά με τη διαφήμιση και αν το επιθυμεί, να προβαίνει και σε άμεση αγορά.

Πέρα από αυτά μια έξυπνη τηλεόραση μπορεί να έχει ενεργό ρόλο σε ένα περιβάλλον έξυπνου σπιτιού που σε καθημερινή βάση οικιακές συσκευές θα συνδέονται ασύρματα στο διαδίκτυο μέσω έξυπνων μικροκυκλωμάτων και θα μπορούν να συλλέγουν και να ανταλλάσσουν δεδομένα.

Στη συνέχεια της παρούσας εργασίας θα παρουσιαστεί μια ενδεικτική βιβλιογραφική ανασκόπηση εφαρμογών-χρήσεων που αφορούν μια έξυπνη τηλεόραση στο οικιακό περιβάλλον αλλά και θέματα ασφάλειας που προκύπτουν.

## 1.2 Επισκόπηση βιβλιογραφίας

### 1<sup>ο</sup> Television as an App:Taming the Device Fragmentation Problem

Το πρώτο white paper μελετάει τη βιομηχανία της συνδρομητικής τηλεόρασης και την ανάγκη αυτών των εταιριών να αναπτύξουν μια κοινή εφαρμογή για όλες τις έξυπνες τηλεοράσεις ανεξαρτήτως κατασκευαστή (**Write-Once-Run-Anywhere Solutions**).

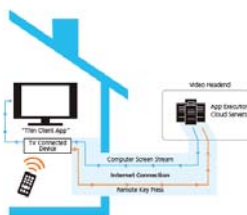
Το μεγάλο πρόβλημα που αντιμετωπίζει αυτή η αγορά είναι ότι αποτελεί καινούργιο χώρο και δεν έχουν διαμορφωθεί ακόμη παγκόσμιες προδιαγραφές. Η αγορά αυτή τη στιγμή είναι κατακερματισμένη με κάθε κατασκευαστή να υλοποιεί διαφορετικά, κάνοντας τη διαδικασία του development μια χρονοβόρα και επίπονη διαδικασία.

Στη συνέχεια το paper μελέτησε κάποιες λύσεις για το πώς θα μπορούσε να επιλυθεί το πρόβλημα της κατακερματισμένης αγοράς, γι αυτό το λόγο παρουσιάζονται κάποιες λύσεις που χρησιμοποιήθηκαν από εταιρίες και μελετώνται ως case study.

Οι τρεις λύσεις που παρουσιάζονται είναι :

- Δημιουργία Developer Kit από τους κατασκευαστές για ανάπτυξη των εφαρμογών (Netflix)
- Χρήση ενός επιπλέον layer του TV Application Layer (TAL) (BBC)
- Universal app platform(tru2way)

Όλες οι λύσεις έχουν κάποια μειονεκτήματα, η λύση που προτείνεται από το paper είναι η ανάπτυξη εφαρμογών στο cloud, η ύπαρξη δηλαδή μιας ελαφριάς έκδοσης στην smart TV που θα συνδέεται με την πλήρη εφαρμογή που βρίσκεται στο cloud. Επίσης το paper αναφέρει χαρακτηριστικά ότι είναι “πολύ λογικό να αναπτύσσονται τα *application* στο *cloud* μιας και όλο και περισσότεροι πάροχοι



μεταφέρουν τα *Video* τους στο *cloud*”.

EIKONA 1 Active Video Networks «Cloud TV» approach

## **2<sup>ο</sup> HTML5 and Companion Web Technologies as a Universal Platform for Interactive Internet TV Advertising**

Το δεύτερο paper μελετάει ένα άλλο χώρο, αυτό της διαφήμισης και πώς οι έξυπνες τηλεοράσεις με τη χρήση των νέων web τεχνολογιών (html5,css3,javascript) μπορούν να δημιουργήσουν διαδραστικές διαφημίσεις με επακόλουθο μια νέα εμπειρία για τον χρήστη. Με τις διαδραστικές διαφημίσεις ο χρήστης παύει να είναι παθητικός δέκτης και αποκτά πια ενεργό ρόλο αλληλεπιδρώντας με τη διαφήμιση και αν το επιθυμεί να προβαίνει ακόμη και σε άμεση αγορά. Βέβαια και σε αυτήν την περίπτωση χρειάζεται οι διαδραστικές διαφημίσεις που φτιάχνονται να μπορούν να είναι συμβατές με πάρα πολλές συσκευές κάνοντας και εδώ επιτακτική την ανάγκη μιας **Universal cross-platform** λύσης. Αυτή τη στιγμή υπάρχει μια πληθώρα από πλατφόρμες διαφόρων κατασκευαστών. Το συγκεκριμένο paper προσπαθεί να γεφυρώσει αυτό το ετερογενές περιβάλλον με χρήση των νέων web τεχνολογιών.

Τα νέα web πρότυπα έχουν ενσωματώσει πολλά χαρακτηριστικά(ενδεικτικά video,audio,typography) για τη δημιουργία ελκυστικών διαφημίσεων με εύκολο τρόπο.

Επιπλέον σε περίπτωση που δεν υποστηρίζεται κάτι από την συσκευή του χρήστη, με χρήση “fallback strategy” μιας βιβλιοθήκης της javascript να έχει τη δυνατότητα να δημιουργήσει το interface της διαφήμισης ανάλογα με τι υποστηρίζεται από τη συσκευή.

Μέσω ενός πειράματος αποδείχτηκε ότι με την υιοθέτηση των τελευταίων web-based τεχνικών μπορούν να δημιουργηθούν **ανεξαρτήτου πλατφόρμας** (cross platform) διαφημίσεις.

Στο paper τονίζεται ότι η χρήση των html5,css3,javascript ανοίγει νέους δρόμους για την δημιουργία νέων μορφών πιο εντυπωσιακών και πιο φιλικών προς το χρήστη διαφημίσεων.



EIKONA 2 The Landing Page / Branded Microsite in HTML5

### 3<sup>ο</sup> Roles of Smart TV in IoT-environments: a Survey

Το τρίτο paper ερευνά κατά πόσο μια έξυπνη τηλεόραση πέρα από τους ρόλους που προαναφέρθηκαν μπορεί να συμμετέχει ενεργά και σε ένα νέο αναπτυσσόμενο περιβάλλον το «Internet of Things».

Το Internet of Things(IoT) είναι ένα περιβάλλον που καθημερινά αντικείμενα (όπως κινητά τηλέφωνα, αυτοκίνητα, οικιακές συσκευές, ρούχα) θα συνδέονται ασύρματα στο διαδίκτυο μέσω έξυπνων μικροκυκλωμάτων και θα μπορούν να συλλέγουν και να ανταλλάσσουν δεδομένα. \*

Μελέτησε τη συμπεριφορά πέντε(5) πολύ γνωστών συσκευών και πώς ανταποκρίνονται σε πέντε(5) ρόλους που θα μπορούσε να έχει μια έξυπνη τηλεόραση σε ένα περιβάλλον IoT.

Οι πέντε ρόλοι με ενδεικτικά παραδείγματα είναι:

1. Information storage: Αποθήκευση ταινιών στην τηλεόραση αντί για σε usb
2. Visualization device: Εμφάνιση της θερμοκρασίας δωματίου στην οθόνη
3. Interaction point: Ρύθμιση της θερμοκρασίας δωματίου
4. Data processor: Ενεργοποίηση μιας άλλης οικιακής συσκευής μια συγκεκριμένη ώρα
5. Data source: Η λειτουργία της ως σύστημα ασφαλείας

Οι πέντε κατασκευαστές είναι :

1. Samsung Smart TV
2. Panasonic Viera Connect
3. LG Smart TV
4. Microsoft Mediaroom
5. Google TV

|           | Storage | Visualization | Interaction | Processing | Source of data |
|-----------|---------|---------------|-------------|------------|----------------|
| Samsung   | Good    | Good          | Good        | Average    | Good           |
| Panasonic | Good    | Good          | Good        | Average    | Good           |
| LG        | Good    | Good          | Good        | Average    | Good           |
| Mediaroom | Average | Bad           | Good        | Average    | Good           |
| Google TV | Good    | Good          | Good        | Good       | Good           |

EIKONA 3

**Good** : η πλατφόρμα ανταποκρίθηκε σε όλα τα πειράματα του συγκεκριμένου ρόλου

**Average**: έχει κάποιους περιορισμούς

**Bad**: είχε μικρή ή καθόλου ανταπόκριση

Από το συγκεκριμένο πείραμα διαπιστώθηκε ότι οι περισσότερες πλατφόρμες ανταποκρίθηκαν με μεγάλη επιτυχία στους διαφορετικούς ρόλους του πειράματος, με την Google TV να ανταποκρίνεται με απόλυτη επιτυχία σε όλους τους ρόλους του πειράματος. Το paper αναφέρει ότι η τεχνολογία των τηλεοράσεων μελλοντικά μπορεί να ανταποκριθεί με επιτυχία και σε άλλα περιβάλλοντα IoT πέραν του οικιακού, όπως το χώρο της υγείας και των μεταφορών.

\* ορισμός από το paper

## 4<sup>ο</sup> Smart TV Hacking: Crash Testing Your Home Entertainment

Με την προσθήκη πολλών χαρακτηριστικών που παλιότερα υπήρχαν μόνο σε υπολογιστές οι σύγχρονες τηλεοράσεις συγκλίνουν τεχνολογικά αρκετά με αυτούς, για αυτό το λόγο κρίνεται απαραίτητο να μελετηθεί η συμπεριφορά της τηλεόρασης σε θέματα ασφαλείας.

Το τελευταίο white paper παρουσιάζει διαφορετικές μορφές επιθέσεων που μπορεί να είναι είτε οι γνωστές επιθέσεις από τους υπολογιστές είτε οι νέες μορφές επιθέσεων εξειδικευμένες στις τηλεοράσεις.

Υπάρχουν δύο μορφές επιθέσεων οι παθητικές και οι ενεργητικές επιθέσεις. Ως παθητικές αναφέρονται, όταν απλά παρακολουθείται ο στόχος(listen) χωρίς να αλλάζονται τα δεδομένα. Σε αυτήν την περίπτωση ο επιτιθέμενος αλληλεπιδρά ελάχιστα με το στόχο. Ενώ αντίθετα οι ενεργητικές αλληλεπιδρούν με το στόχο τους και σαν σκοπό έχουν να τον βλάψουν. Το συγκεκριμένο paper ασχολείται με τις ενεργητικές επιθέσεις.

Κάποια ενδεικτικά σενάρια επιθέσεων που μπορούν να συμβούν και να εκτεθεί μια τηλεόραση είναι:

- Denial Of Service attacks (DoS) με αποτέλεσμα η τηλεόραση-στόχος να “κρυστάλλει”
- Είναι εκτεθειμένες σε Malicious code που μπορεί να τρέχει κρυφά στην τηλεόραση
- Υποκλοπή δεδομένων, όπως στοιχεία πιστωτικής που αποθηκεύονται στο υπολογιστή
- Ιοί από μολυσμένα usb

Το paper πραγματοποίησε μια έρευνα χρησιμοποιώντας fuzzing tool\*\* για να διαπιστώσει πώς ανταποκρίνονται οι τηλεοράσεις στις διάφορες επιθέσεις.

### Πείραμα Μεθοδολογία

Όλες οι τηλεοράσεις που χρησιμοποιήθηκαν στο πείραμα ήταν ενημερωμένες με το τελευταίο firmware του κατασκευαστή. Όλες οι υπηρεσίες των τηλεοράσεων ήταν ενεργοποιημένες για να καλυφθούν από το πείραμα όλες οι πιθανές περιπτώσεις.

Πραγματοποιήθηκε σκανάρισμα(scan) όλων των ανοικτών πορτών προκειμένου να βρεθούν όλα τα τρωτά σημεία που θα μπορούσαν να είναι πιθανοί στόχοι μιας επίθεσης(attack surface).Μελετήθηκαν όλα τα documentation και ελέγχθηκαν όλα τα πρωτόκολλα που χρησιμοποιούνται.

Από το πείραμα διαπιστώθηκε ότι και τα παλιά πρωτόκολλα, όπως το IPv4, αλλά και πρωτόκολλα που χρησιμοποιούνται από τις τηλεοράσεις αποκλειστικά απέτυχαν σε κάποια test.

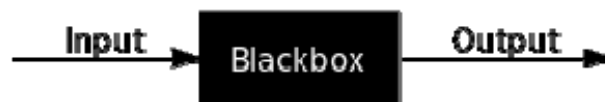
| Protocol | Vendor 1 | Vendor 2 | Vendor 3 | Vendor 4 | Vendor 5 | Vendor 6 |
|----------|----------|----------|----------|----------|----------|----------|
| IPv4     | pass     | fail     | fail     | pass     | pass     | fail     |
| DVB      | fail     | fail     | fail     | fail     | fail     | fail     |
| UPnP     | n/a      | fail     | pass     | n/a      | n/a      | fail     |
| Images   | pass     | fail     | fail     | n/a      | n/a      | fail     |
| Audio    | pass     | pass     | n/a      | n/a      | n/a      | pass     |
| Video    | fail     | fail     | pass     | fail     | fail     | fail     |

• pass - survived  
• fail - vulnerable  
• n/a - not available for testing

EIKONA 4

Μετά από το test που πραγματοποιήθηκε διαπιστώθηκε ότι πέρα από βασικούς μηχανισμούς ασφαλείας που πρέπει να εφαρμόζονται για να ισχύουν οι βασικές έννοιες της ασφάλειας(Confidentiality,Integrity,Authentication) κρίνεται απαραίτητο να υπάρχουν και επιπρόσθετοι μηχανισμοί ασφαλείας, όπως fuzzing test.

Οι τεχνικές Black-Box Testing (fuzzing test) πρέπει να εφαρμόζονται από το κατασκευαστή σε όλο το κύκλο της ανάπτυξης των τηλεοράσεων, πριν βγουν προς διάθεση στο καταναλωτικό κοινό. Οι κατασκευαστές θα πρέπει να είναι σε θέση πολύ εύκολα και γρήγορα να διαθέτουν updates.



EIKONA 5 “Diagram Black-Box Testing”\*\*\*

\*\*Fuzz testing [http://en.wikipedia.org/wiki/Fuzz\\_testing](http://en.wikipedia.org/wiki/Fuzz_testing)

\*\*\* Diagram Black-Box Testing [http://en.wikipedia.org/wiki/Black-box\\_testing](http://en.wikipedia.org/wiki/Black-box_testing)

### 1.3 Έρευνες Nielsen - Eurostat

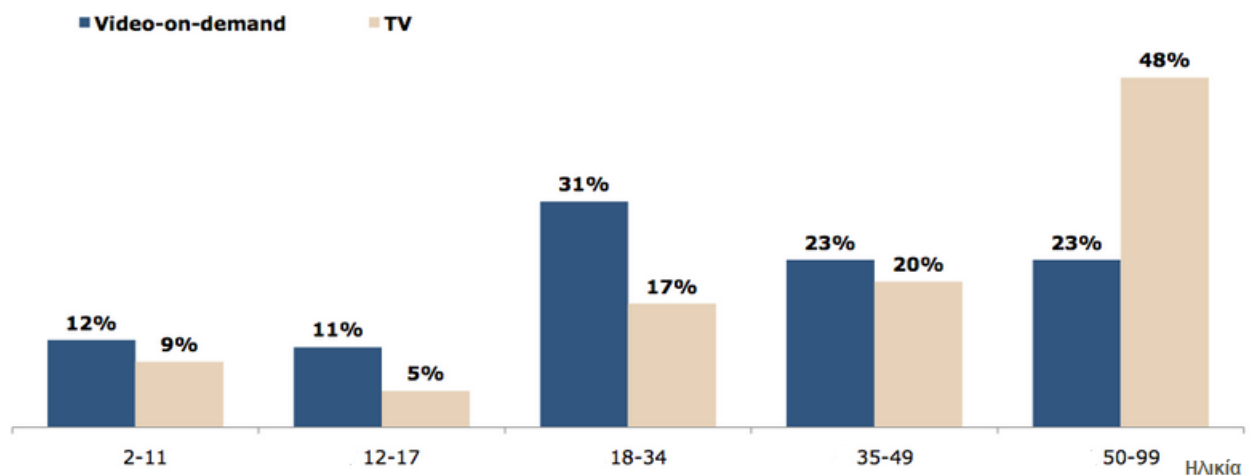
Στην συνέχεια παρουσιάζονται ορισμένα στοιχεία των ατόμων που κάνουν χρήση των VoD υπηρεσιών που πραγματοποιήθηκε από τον Nielsen στην Αγγλία και από την Eurostat που αφορούν τον πολιτιστικό τομέα στις χώρες της Ευρωπαϊκής ένωσης.

#### 1.3.1 Nielsen: Χρήση VoD



ΕΙΚΟΝΑ 6

Σύμφωνα με έρευνα που πραγματοποιήθηκε από το Nielsen τον Σεπτέμβριο του 2013 σε σπίτια στην Αγγλία παρουσιάζεται η χρήση των υπηρεσιών VOD έναντι της κλασικής τηλεόρασης ανά ηλικία



ΕΙΚΟΝΑ 7, Πηγή: Nielsen, 2013

| Κατηγορία   | Έχουν παιδιά | Δεν έχουν παιδιά |
|-------------|--------------|------------------|
| Χρήστες VOD | 54%          | 46%              |
| Χρήστες TV  | 40%          | 60%              |

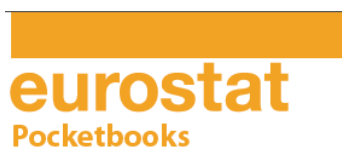
ΕΙΚΟΝΑ 8

| Κατηγορία   | Εισόδημα >100.000δολάρια |
|-------------|--------------------------|
| Χρήστες VOD | 31%                      |
| Χρήστες TV  | 21%                      |

ΕΙΚΟΝΑ 9

Από την έρευνα αυτή προκύπτει ότι οι χρήστες είναι κυρίως ηλικίας 18-34 με παιδιά .

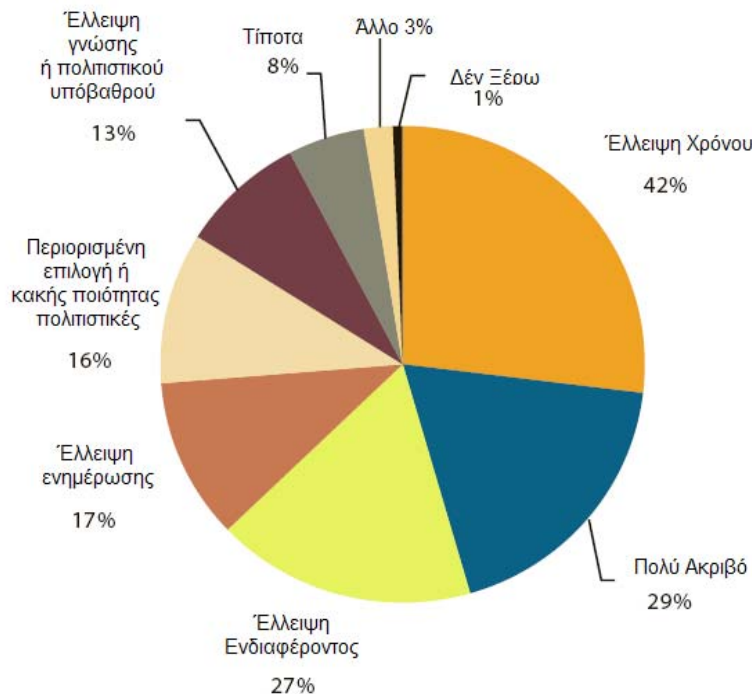
### 1.3.2 Eurostat: «στατιστικά στον πολιτιστικό τομέα» ΕΕ-27



ΕΙΚΟΝΑ 10

Ακολουθούν έρευνες που πραγματοποιήθηκαν από την eurostat το 2011 για την συμμετοχή των μελών της Ευρωπαϊκής Ένωσης σε δραστηριότητες του πολιτισμού.

**Ερώτηση: «Μερικές φορές οι άνθρωποι βρίσκουν δύσκολο να έχουν πρόσβαση στον πολιτισμό ή να λάβουν μέρος σε πολιτιστικές δραστηριότητες. Ποιο από τα ακόλουθα, εάν υπάρχουν, είναι τα κύρια εμπόδια για σας;»**



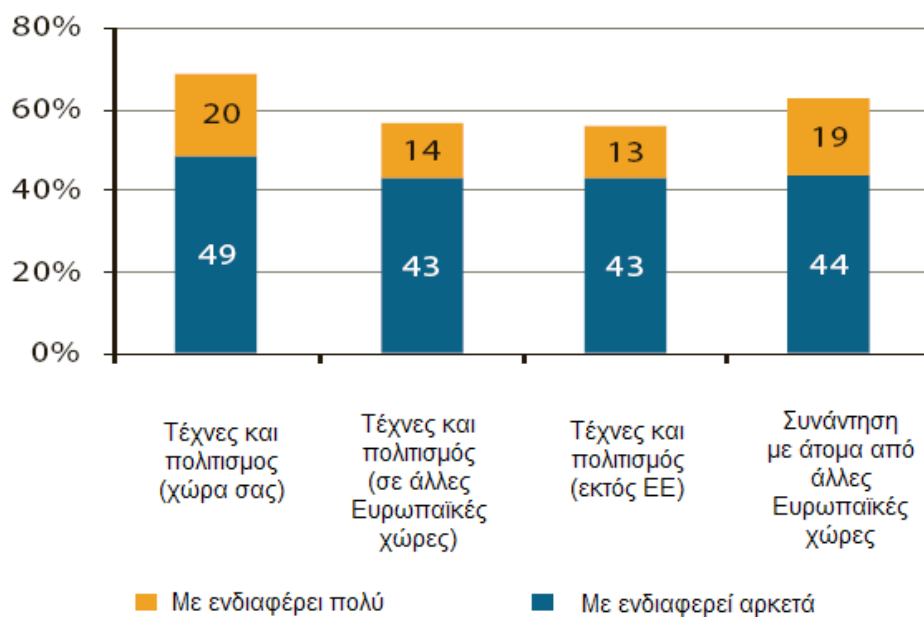
ΕΙΚΟΝΑ 11

Πηγή: Ευρωβαρόμετρο 67.1, 2007

**Ερώτηση: Σε ποιο βαθμό τα ακόλουθα θέματα σας ενδιαφέρουν;**

1. πολύ
2. αρκετά
3. όχι πολύ ενδιαφέρονται
4. δεν ενδιαφέρονται καθόλου;

**Πόσο θα λέγατε ότι σας ενδιαφέρει να συναντήσετε άτομα από άλλες Ευρωπαϊκές χώρες;**



ΕΙΚΟΝΑ 12, Πηγή: Ευρωβαρόμετρο 67.1, 2007

Από τις έρευνες παρατηρούμε ότι οι 3 κυριότεροι αποτρεπτικοί παράγοντες για συμμετοχή και παρακολούθηση πολιτιστικών δραστηριοτήτων είναι :

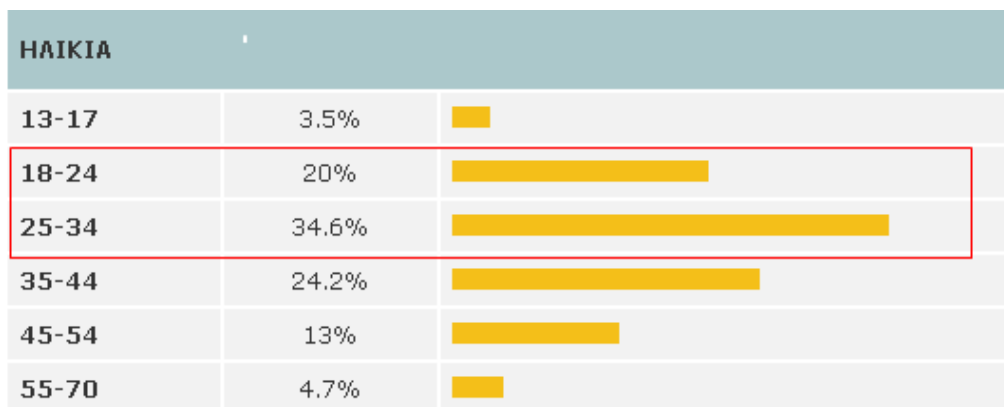
- η έλλειψη χρόνου,
- το κόστος
- και η απόσταση από το φυσικό τόπο διεξαγωγής.

### 1.3.3 Αθηνόγραμμα

#### Σύντομο προφίλ

Το αθηνόγραμμα δημιουργήθηκε για να αποτελέσει ένα ολοκληρωμένο οδηγό πώλης. Από το 1996 μέχρι σήμερα βρίσκεται ανάμεσα στα πρώτα από άποψη επισκεψιμότητας sites του ελληνικού Internet, κατέχοντας σταθερά την κορυφή μεταξύ των sites που αφορούν την ψυχαγωγία και τη διασκέδαση.

Σύμφωνα με έρευνες που πραγματοποιεί το ίδιο το αθηνόγραμμα.gr για το προφίλ των χρηστών του παρατηρούμε από τον πίνακα της έρευνας που ακολουθεί ότι τα άτομα που επισκέπτονται περισσότερο την σελίδα για να ενημερωθούν είναι ηλικίας 18 έως 34. Τέλος το μεγαλύτερο ποσοστό των ατόμων αυτών αναζητούν πληροφορίες στην κατηγορία “θέατρο”.



ΕΙΚΟΝΑ

Έτος:2013

|                   | Ιανουάριος | Φεβρουάριος | Μάρτιος    | Απρίλιος  | Μάιος     | Ιούνιος   | Ιούλιος   | Αύγουστος | Σεπτέμβριος | Οκτώβριος  | Νοέμβριος  | Δεκέμβριος |
|-------------------|------------|-------------|------------|-----------|-----------|-----------|-----------|-----------|-------------|------------|------------|------------|
| Unique Visitors   | 1.080.426  | 963.577     | 981.680    | 835.422   | 740.213   | 778.290   | 821.499   | 744.030   | 901.351     | 947.142    | 959.012    | 1.207.930  |
| Visits            | 2.550.744  | 2.192.671   | 2.184.932  | 1.813.045 | 1.527.409 | 1.623.426 | 1.713.298 | 1.473.965 | 1.825.153   | 1.949.183  | 1.986.187  | 2.437.987  |
| Total Page Views  | 13.982.769 | 11.797.559  | 11.385.764 | 8.426.071 | 6.693.518 | 7.464.195 | 8.407.219 | 7.390.985 | 8.408.613   | 10.133.053 | 10.375.269 | 11.617.701 |
| Visits            | 581.863    | 491.386     | 448.255    | 349.245   | 169.295   | 141.233   | 172.559   | 125.019   | 189.016     | 329.549    | 463.544    | 517.594    |
| Θέατρο Page Views | 3.249.643  | 2.593.842   | 2.253.042  | 1.547.936 | 593.347   | 495.114   | 554.659   | 386.732   | 669.303     | 1.682.366  | 2.408.375  | 2.531.274  |

Έτος:2012

|                   | Ιανουάριος | Φεβρουάριος | Μάρτιος    | Απρίλιος  | Μάιος     | Ιούνιος   | Ιούλιος   | Αύγουστος | Σεπτέμβριος | Οκτώβριος  | Νοέμβριος  | Δεκέμβριος |
|-------------------|------------|-------------|------------|-----------|-----------|-----------|-----------|-----------|-------------|------------|------------|------------|
| Unique Visitors   | 925.550    | 802.425     | 804.258    | 661.399   | 607.329   | 596.758   | 635.385   | 573.567   | 732.532     | 846.915    | 916.501    | 1.061.026  |
| Visits            | 2.204.986  | 1.851.140   | 1.869.864  | 1.436.094 | 1.324.531 | 1.281.923 | 1.359.282 | 1.180.289 | 1.596.294   | 1.903.014  | 2.107.153  | 2.363.498  |
| Total Page Views  | 13.509.205 | 10.721.187  | 10.655.159 | 7.352.365 | 6.310.851 | 6.708.495 | 7.434.808 | 6.824.509 | 8.117.455   | 10.622.785 | 11.909.497 | 13.088.178 |
| Visits            | 506.264    | 366.672     | 400.649    | 225.127   | 161.640   | 130.819   | 154.893   | 112.756   | 184.250     | 352.694    | 472.061    | 487.927    |
| Θέατρο Page Views | 3.371.505  | 2.039.629   | 2.118.595  | 1.063.981 | 651.873   | 457.843   | 501.849   | 325.680   | 600.991     | 1.953.731  | 2.889.297  | 2.958.470  |

ΕΙΚΟΝΑ 13

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ - ΕΝΝΟΙΕΣ

Το κεφάλαιο που ακολουθεί έχει ως σκοπό να παρουσιάσει τις βασικές έννοιες που απαιτούνται για την κατανόηση της υλοποίησης.

Αναλυτικότερα το κεφάλαιο αυτό χωρίζεται σε δυο ενότητες, η πρώτη ενότητα αφορά ορισμούς σχετικά με την εφαρμογή, τα πρωτοκόλλα που απαιτούνται και τεχνολογίες που υποστηρίζονται από μια έξυπνη τηλεόραση.

Η δεύτερη ενότητα αναφέρεται στις διαφορετικές κατηγορίες κατασκευαστών που υπάρχουν. Τέλος θα γίνει παρουσίαση ενός ενδεικτικού παραδείγματος της κάθε κατηγορίας.

### 2.1 Ορισμοί

**Multimedia(πολυμέσο):** αναφέρεται στο περιεχόμενο που μπορεί να περιλαμβάνει ένα συνδυασμό κειμένου, ήχου, εικόνας, κινουμένων σχεδίων, βίντεο ή διαδραστικών μέσων.

**Streaming media:** βίντεο ή ήχος που στέλνεται σε συμπιεσμένη μορφή μέσω του διαδικτύου και μπορεί να γίνει η αναπαραγωγή του αμέσως, αντί να αποθηκεύεται στον σκληρό δίσκο.

**Streaming video:** βίντεο του οποίου το περιεχόμενο αποστέλλεται σε συμπιεσμένη μορφή μέσω του διαδικτύου και εμφανίζεται στον θεατή σε πραγματικό χρόνο. Με streaming video ή streaming media, ένας χρήστης στο διαδίκτυο δε χρειάζεται να περιμένει να γίνει η λήψη του αρχείου για να εκτελεστεί. Αντί αυτού το media αποστέλλεται με μια συνεχή ροή δεδομένων και εκτελείται, όπως φτάνει. Ο χρήστης χρειάζεται ένα λογισμικό για την εκτέλεση του βίντεο, το οποίο αποσυμπιέζει και στέλνει τα δεδομένα στην οθόνη και τον ήχο στα ηχεία. Ένα λογισμικό media player μπορεί να είναι είτε ένα αναπόσπαστο τμήμα ενός προγράμματος περιήγησης browser, ή μπορεί να ληφθεί downloaded από την ιστοσελίδα του κατασκευαστή του λογισμικού.

**Progressive download(προοδευτική λήψη):** μεταφορά των ψηφιακών αρχείων-πολυμέσων από ένα διακομιστή σε έναν χρήστη χρησιμοποιώντας συνήθως το πρωτόκολλο HTTP, όταν ενεργοποιείται από έναν υπολογιστή. Η προοδευτική λήψη επιτρέπει στον χρήστη την πρόσβαση στο περιεχόμενο, πριν η μεταφορά των δεδομένων ολοκληρωθεί. Η εμπειρία με την προοδευτική λήψη είναι παρόμοια με αυτή που προσφέρεται από το streaming, πολλοί χρήστες δεν μπορούν να διακρίνουν διαφορές μεταξύ των δυο. Ωστόσο με το ληφθέν περιεχόμενο δεν είναι δυνατή η γρήγορη

προώθηση προς τα εμπρός οποιουδήποτε τμήματος του αρχείου, μέχρι αυτό να αποθηκευτεί στη συσκευή του χρήστη. Παρά το γεγονός ότι υπάρχει πάντα κάποια καθυστέρηση που σχετίζεται με την αναπαραγωγή του ληφθέντος περιεχομένου η καθυστέρηση μπορεί να είναι τόσο σύντομη που δεν εντοπίζεται.

## **Τύποι Streaming**

**Vod/AVod** (βίντεο κατόπιν ζήτησης ή ήχος κατόπιν ζήτησης): συστήματα τα οποία επιτρέπουν στους χρήστες να επιλέξουν και να παρακολουθήσουν/ακούσουν το βίντεο ή ήχο κατόπιν ζήτησης(on demand).

**True streaming:** αναφέρεται σε τεχνολογίες που ταιριάζουν το εύρος ζώνης (bandwidth) του σήματος του μέσου με τη σύνδεση του χρήστη έτσι ώστε το μέσο να εμφανίζεται σε πραγματικό χρόνο. Η λέξη true προστίθεται για την διαφοροποίηση αυτού του είδους streaming από το HTTP streaming. Εξειδικευμένοι διακομιστές media καθώς επίσης και πρωτόκολλα streaming, όπως rtsp είναι απαραίτητα για πραγματικό streaming

## **Πρωτόκολλα Streaming**

**Real-time Transport Protocol(RTP control protocol):** ορίζει μια τυποποιημένη μορφή πακέτου για την παράδοση/μεταφορά ήχου και βίντεο μέσω δικτύων ip. Το RTP χρησιμοποιείται στα συστήματα επικοινωνίας και διασκέδασης που περιλαμβάνουν streaming media, όπως τηλεφωνία, βίντεο-κλήσεις, τηλεοπτικές υπηρεσίες και διαδικτυακές push to talk υπηρεσίες. πρωτόκολλο μεταφοράς.

**Real-Time Streaming Protocol(RTSP):** πρωτόκολλο ελέγχου δικτύου σχεδιασμένο για χρήση σε συστήματα διασκέδασης και επικοινωνιών για τον έλεγχο media streaming διακομιστών(media server). Το πρωτόκολλο χρησιμοποιείται για τον καθορισμό και τον έλεγχο των συνεδρίων(Session) media μεταξύ των τελικών σημείων.

**RTP Control Protocol (RTCP):** συγγενικό πρωτόκολλο του πραγματικού χρόνου RTP.

**Adaptive Bitrate streaming:** μια τεχνική που χρησιμοποιείται στο streaming πολυμέσων σε δίκτυα ηλεκτρονικών υπολογιστών. Ενώ στο παρελθόν οι περισσότερες τεχνολογίες video streaming χρησιμοποιούσαν πρωτόκολλα RTP με RTSP σήμερα οι τεχνολογίες streaming είναι κατά αποκλειστικότητα βασισμένες σε HTTP και σχεδιασμένες να λειτουργούν αποτελεσματικά σε

μεγάλα κατανεμημένα HTTP δίκτυα, όπως το διαδίκτυο. Λειτουργεί ανιχνεύοντας το εύρος ζώνης(bandwidth) του χρήστη και της ικανότητας της κεντρικής μονάδας επεξεργασίας(cpu) σε πραγματικό χρόνο, καθώς και να προσαρμόζει την ποιότητα του βίντεο stream αναλόγως.

### **Εφαρμογές τεχνικής Adaptive Bitrate streaming**

**HTTP Live Streaming (HLS):** ένα HTTP πρωτόκολλο επικοινωνίας (HTTP-based) media streaming που εφαρμόζεται από την apple inc, ως μέρος του QuickTime και του λογισμικού ios.

**HTTP Dynamic Streaming (HDS):** Το HDS δίνει τη δυνατότητα για μετάδοση on demand και live adaptive bitrate video που βασίζονται σε mp4 media μέσω των κοινών http συνδέσεων. HDS επιτρέπει την χρήση των υφιστάμενων υποδομών cache και παρέχει τα εργαλεία για την ολοκλήρωση της προετοιμασίας του περιεχομένου σε υπάρχουσες εργασίες κωδικοποίησης. Οι τεχνικές κωδικοποίησης βίντεο και τα πρότυπα βασίζονται σε ένα σύνολο αρχών που μειώνουν τον πλεονασμό σε ψηφιακό βίντεο.

**Dynamic adaptive streaming μέσω http (Dash):** Το Dash είναι επίσης γνωστό ως mpeg-Dash επιτρέπει υψηλής ποιότητας streaming πολυμέσων μέσω του διαδικτύου που παραδίδονται από τους συμβατικούς HTTP web servers.(διακομιστές).

**Smooth streaming:** Το smooth streaming είναι μια ISS media services επέκταση που επιτρέπει την προσαρμοστική ροή των media σε Silverlight και σε άλλους clients μέσω HTTP. Smoothes streaming προσφέρει μια υψηλής ποιότητας εμπειρία θέασης που κλιμακώνεται μαζικά στα δίκτυα διανομής περιεχομένου πραγματοποιώντας μια αληθινή εμπειρία HD 1080p.

### **Τεχνολογίες που υποστηρίζει μια smart TV**

**Web 2.0:** Το web2.0 χρησιμοποιείται για να περιγράψει τη νέα γενιά του Παγκόσμιου Ιστού η οποία βασίζεται στην όλο και μεγαλύτερη δυνατότητα των χρηστών του Διαδικτύου να μοιράζονται πληροφορίες και να συνεργάζονται online. Αυτή η νέα γενιά είναι μια δυναμική διαδικτυακή πλατφόρμα στην οποία μπορούν να αλληλεπιδρούν χρήστες χωρίς εξειδικευμένες γνώσεις σε θέματα υπολογιστών και δικτύων.

**Semantic HTML:** χρησιμοποιεί Html για να ενισχύσει την κατασκευαστική structural έννοια. Χρησιμοποιώντας ετικέτες, ονόματα κατηγορίας και ταυτότητες ενισχύουν την έννοια του περιεχομένου των ετικετών.

**HTML5:** γλώσσα σήμανσης που χρησιμοποιείται για την κατασκευή και παρουσίαση περιεχομένου στο διαδίκτυο. Είναι η πέμπτη αναθεώρηση του προτύπου HTML και προορίζεται να αντικαταστήσει τις HTML 4.01, XHTML 1.0, και DOM Level 2 HTML. Οι κύριοι στόχοι της είναι η βελτιστοποίηση της γλώσσας για να υποστηρίξει πολυμέσα καθιστώντας αυτά πιο ευκολοδιάβαστα για τους ανθρώπους και κατανοητά από υπολογιστές και συσκευές. Η HTML5 περιλαμβάνει ένα καινούργιο σετ από σημασιολογικές semantic ετικέτες.

**Cascading style sheets(CSS):** μια γλώσσα που χρησιμοποιείται για την περιγραφή των παρουσιαστικών σημασιολογικών, την εμφάνιση και μορφοποίηση ενός έγγραφου γραμμένο σε μια γλώσσα σήμανσης.

**CSS3 :** το πιο πρόσφατο πρότυπο για css

**DHTML:** συνδυασμός των τεχνολογιών HTML, JavaScript CSS & DOM. Χρησιμοποιώντας αυτές τις τεχνολογίες οι ιστοσελίδες μπορούν να αλλάζουν κατά τη χρήση τους και να αλληλεπιδρούν με τους χρήστες χωρίς ανανέωση της σελίδας .

**Asynchronous JavaScript and XML(AJAX) :** ένας συνδυασμός web τεχνολογιών, είναι μια τεχνική ανταλλαγής δεδομένων με ένα διακομιστή, και η ενημέρωση τμημάτων μιας σελίδας web - χωρίς επαναφόρτωση ολόκληρης της σελίδας.

## **Κατηγορίες τηλεοράσεων**

**Interactive Television(Διαδραστική τηλεόραση):** μια μορφή της σύγκλισης των μέσων, προσθέτοντας υπηρεσίες δεδομένων στην παραδοσιακή τηλεόραση.

**Internet Television:** ψηφιακή μετάδοση τηλεοπτικού περιεχομένου μέσω διαδικτύου.

**Smart Tv:** περιγράφει την τάση για την ενσωμάτωση του διαδικτύου και web 2.0 χαρακτηριστικών σε τηλεοράσεις και σε αποκωδικοποιητές καθώς επίσης των τεχνολογικών συγκλίσεων μεταξύ υπολογιστών και τηλεοράσεων και αποκωδικοποιητών. Οι smart tv υποστηρίζουν επιπλέον

Ethernet, Wi-Fi, USB port, Bluetooth και κάρτες μνήμης flash από ψηφιακές φωτογραφικές μηχανές, εκτός από τα κλασικά ομοαξονικά, HDMI καλώδια και άλλες συνδέσεις audio-video

**IPTV:** σύστημα που χρησιμοποιείται για την παροχή ψηφιακών τηλεοπτικών υπηρεσιών στους εγγεγραμμένους χρήστες για το σύστημα αυτό. Η μετάδοση της ψηφιακής τηλεόρασης έγινε εφικτή χρησιμοποιώντας πρωτόκολλο διαδικτύου(IP) με χρήση broadband σύνδεση συνήθως σε διαχειριζόμενο δίκτυο παρά σε δημόσιο διαδίκτυο για να διατηρηθεί η ποιότητα των υπηρεσιών.

**Digital media players (DMP):** συσκευή για κατ'οίκον ψυχαγωγία. Έχει την δυνατότητα να συνδεθεί σε ένα οικιακό δίκτυο για streaming αρχείων ψηφιακών μέσων (όπως μουσική, φωτογραφίες ή βίντεο) από έναν προσωπικό υπολογιστή ή από media server.

## **Πρωτόκολλα Smart Tv**

**DVB:** ένα σύνολο διεθνώς αποδεκτών ανοικτών προτύπων για την ψηφιακή τηλεόραση.

## **Οργανισμοί**

**Digital Living Network Alliance(DLNA):** μη κερδοσκοπικός οργανισμός που αποτελείται από διεθνείς επιχειρήσεις, οι οποίες πέτυχαν την εύκολη σύνδεση μεταξύ συσκευών, όπως για παράδειγμα του υπολογιστή, του Blu-ray Player, του κινητού τηλεφώνου και της τηλεόρασης. Με χρήση της τεχνολογίας DLNA μπορεί να πραγματοποιηθεί η ανταλλαγή ψηφιακών αρχείων μεταξύ διαφόρων ηλεκτρονικών οικιακών συσκευών.

## **Νομικά**

**Digital Rights Management (DRM):** αγγλικό ακρωνύμιο για τη διαχείριση ψηφιακών δικαιωμάτων. Το DRM είναι μια τεχνολογία που χρησιμοποιείται από τις υπηρεσίες παροχής περιεχομένου, όπως τα ηλεκτρονικά καταστήματα, για να ελέγχουν τον τρόπο με τον οποίο χρησιμοποιούνται και διανέμονται τα αρχεία μουσικής και βίντεο που λαμβάνεται. Τα ηλεκτρονικά καταστήματα πωλούν και νοικιάζουν τραγούδια και ταινίες στα οποία έχει εφαρμοστεί η τεχνολογία DRM.

## 2.2 Κατασκευαστές

Σε αυτή την ενότητα θα παρουσιαστούν οι διάφορες κατηγορίες κατασκευαστών που αφορούν σε μια έξυπνη τηλεόραση ή στην χρήση κάποιων χαρακτηριστικών της .

Η πρώτη κατηγορία είναι αυτοί που παραδοσιακά κατασκεύαζαν τηλεοράσεις, αναπτύσσοντας software, hardware δικό τους.

Η ακόλουθη λίστα παρουσιάζει τους γνωστότερους κατασκευαστές έξυπνων τηλεοράσεων

- Samsung Smart TV
- LG
- Panasonic
- Philips
- Toshiba

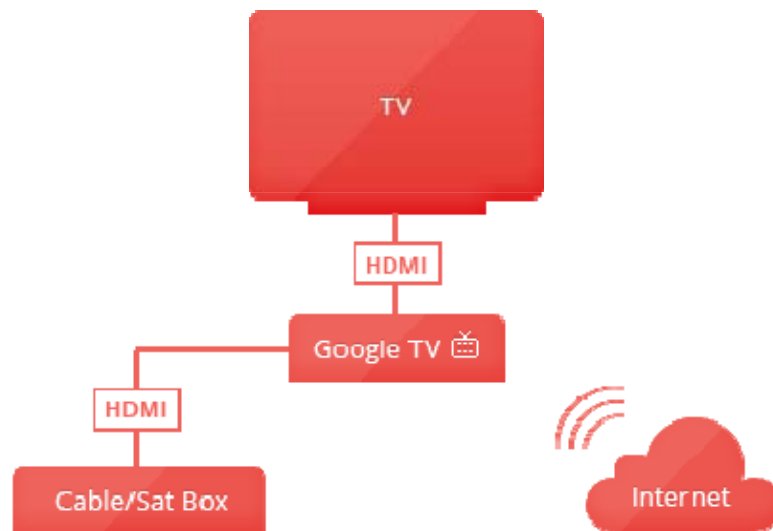
Η δεύτερη κατηγορία είναι μια νέα κατηγορία κατασκευαστών που δεν υπήρχαν παραδοσιακά στο χώρο των τηλεοράσεων αλλά σε συνεργασία με κατασκευαστές hardware έχουν αναπτύξει μια πλατφόρμα .

Χαρακτηριστικό παράδειγμα : Google TV - Android TV

Google TV - Android TV: είναι μια πλατφόρμα για έξυπνη τηλεόραση από το Google που συν-αναπτύχθηκε από την Intel, τη Sony και τη Logitech τον Οκτώβριο του 2010, με επίσημες υποστηριζόμενες συσκευές αρχικά από τη Sony και τη Logitech. Η Google TV - Android TV ενσωματώνει το λειτουργικό σύστημα Android και το Google Chrome για web browser .

Υπάρχουν δυο τρόποι σύνδεσης για την Google TV - Android TV

## 1.Χρήση αποκωδικοποιητή στην κοινή τηλεόραση



EIKONA 14

## 2.Χρήση υποστηριζόμενης τηλεόρασης



EIKONA 15

Η Τρίτη κατηγορία είναι οι **IPTV** και αφορά αποκωδικοποιητές οι οποίοι δίνουν τη δυνατότητα στις απλές τηλεοράσεις να κάνουν χρήση κάποιων υπηρεσιών κυρίως αναπαραγωγής πολυμέσων από κάποια συνδρομητική υπηρεσία .

Χαρακτηριστικά παραδείγματα:

- Mediaroom,
- IPTV on Xbox 360
- Nova
- OTE TV

Mediaroom είναι μια πλατφόρμα IPTV, αρχικά αναπτύχθηκε από τη Microsoft και στη συνέχεια την απέκτησε η Ericsson.

Τέλος, με χρήση Digital media players χαρακτηριστικό παράδειγμα Apple TV

Apple TV ή iTV που αναπτύχθηκε από την Apple. Πρόκειται για μια μικρή συσκευή δικτύου και συσκευή ψυχαγωγίας, σχεδιάστηκε για να παίζει το ψηφιακό περιεχόμενο από το iTunes Store, Netflix, Hulu Plus, το YouTube και το Vevo, μαζί με την τηλεόραση.

## ΚΕΦΑΛΑΙΟ 3<sup>ο</sup> ΠΕΡΙΓΡΑΦΗ ΥΛΟΠΟΙΗΣΗΣ

### 3.1 Περιγραφή των εργαλείων



ΕΙΚΟΝΑ 16

Για την υλοποίηση της εφαρμογής επιλέχτηκε η πλατφόρμα της Samsung, ως συμβατός κατασκευαστής.

Η Samsung υποστηρίζει διάφορες τεχνολογίες για την ανάπτυξη των εφαρμογών της, ενδεικτικά παρουσιάζονται μερικές από αυτές :

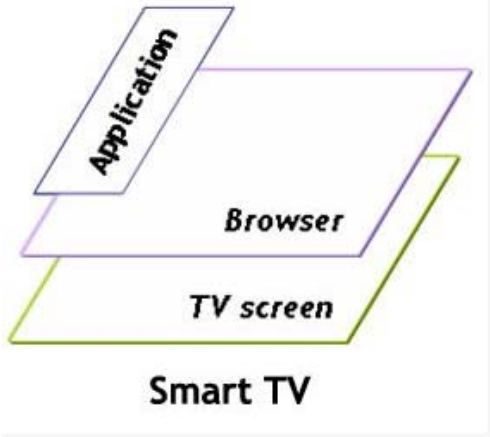
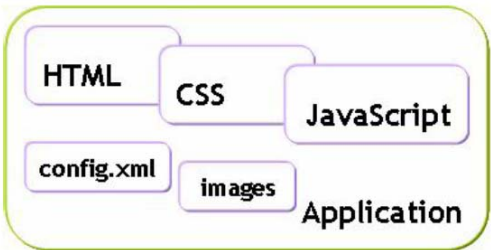
- Web πρότυπα web framework (HTML5-CSS3)
- DOM
- Javascript
- AIR
- Πρωτόκολλα stream RTP/RTSP

Για την ανάπτυξη των εφαρμογών της έχει δημιουργήσει το δικό SDK, το **Samsung Smart TV SDK**.

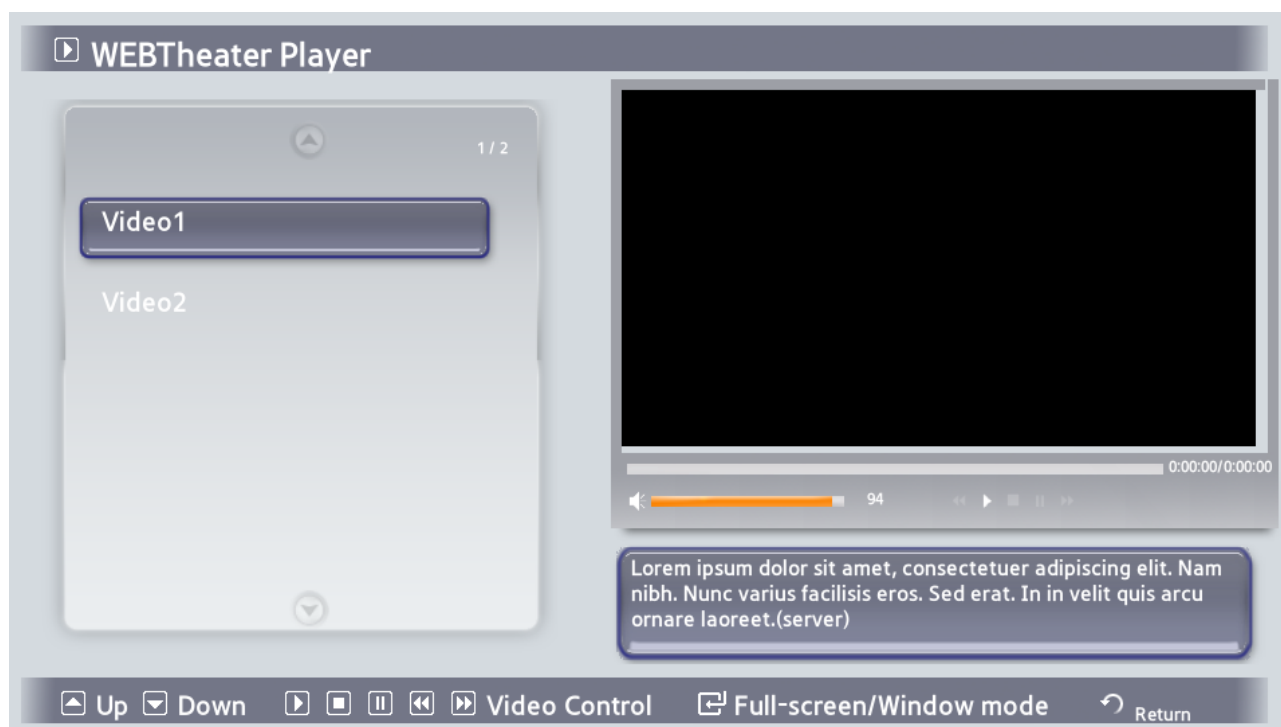
Το SDK περιλαμβάνει:

- eclipse editor για τη δημιουργία των αρχείων
- ένα περιβάλλον δοκιμής της εφαρμογής emulator
- και τέλος πακετάρισμα της εφαρμογής

Η εφαρμογή είναι συμβατή μόνο με έξυπνες τηλεοράσεις της Samsung, αφού δεν υπάρχει ακόμη κοινή πλατφόρμα ανάπτυξης όλων των κατασκευαστών, όπως στα έξυπνα κινητά .

|                                                                                   |                                                                                    |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
|  |  |
| <p>EIKONA 17. αρχιτεκτονική μια έξυπνης τηλεόρασης</p>                            | <p>EIKONA 18. Δομή εφαρμογής</p>                                                   |

### 3.2 Δομή εφαρμογής - Περιγραφή



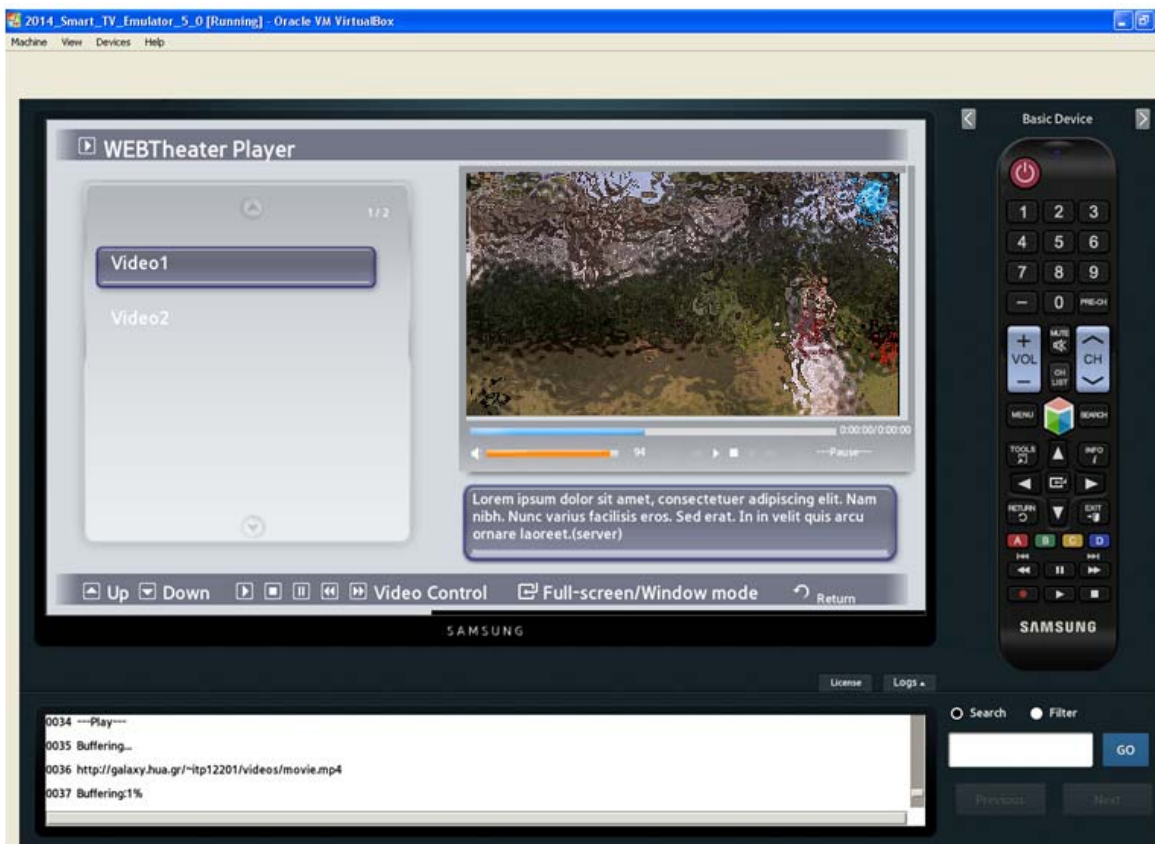
EIKONA 19

| Γενικά χαρακτηριστικά<br>εφαρμογής        |
|-------------------------------------------|
| Χρήση web προτύπων<br>HTML,CSS,Javascript |
| Επικοινωνία με τον server:Ajax            |
| Ανάλυση:960x540 (bpp)                     |
| Συμβατότητα μοντέλα:2013-2014             |
| SDK: 4.5, 5                               |

EIKONA 20

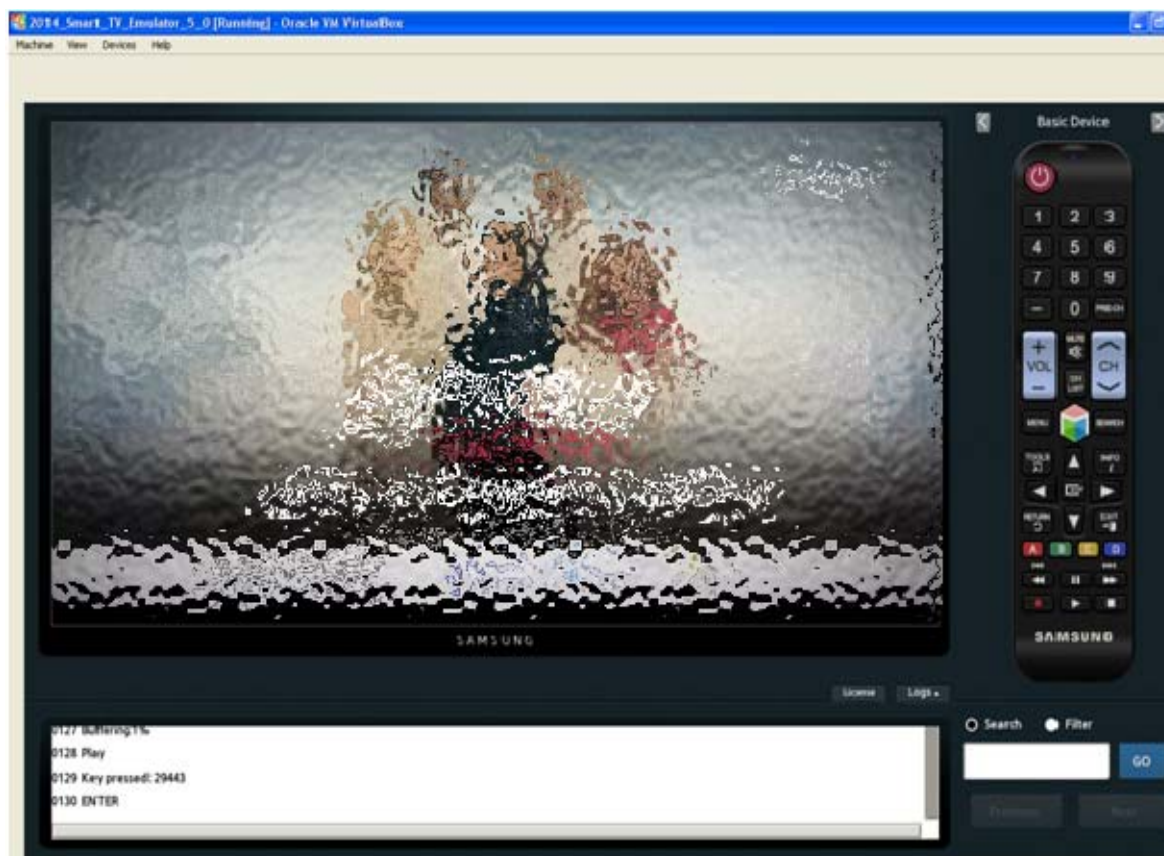
Η εφαρμογή έχει δοκιμαστεί και δουλεύει χωρίς πρόβλημα για τα μοντέλα της Samsung 2013-2014

Μοντέλο 2014



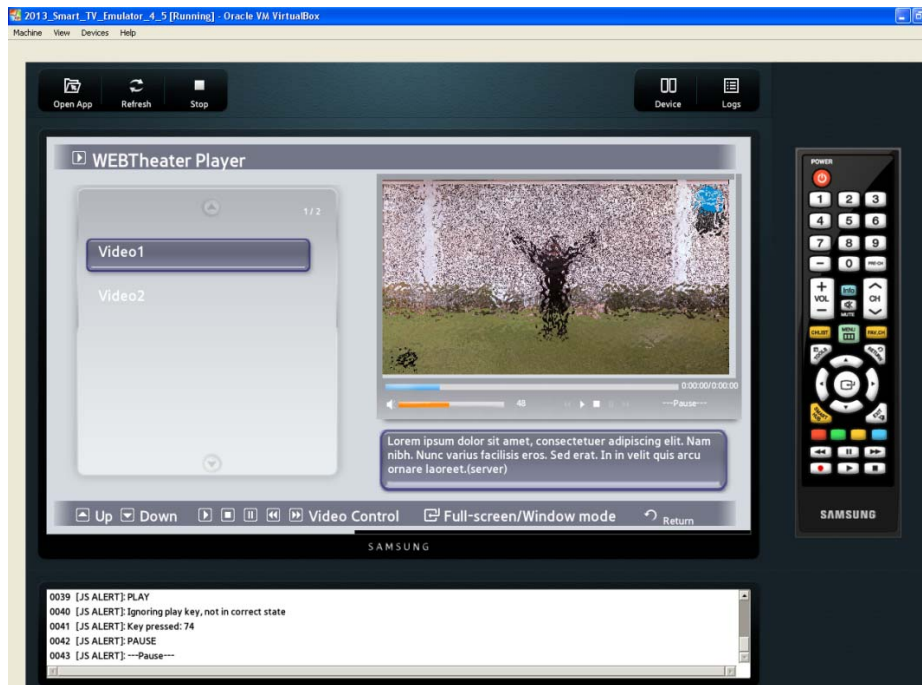
EIKONA 21

Μοντέλο 2014 \_ full screen



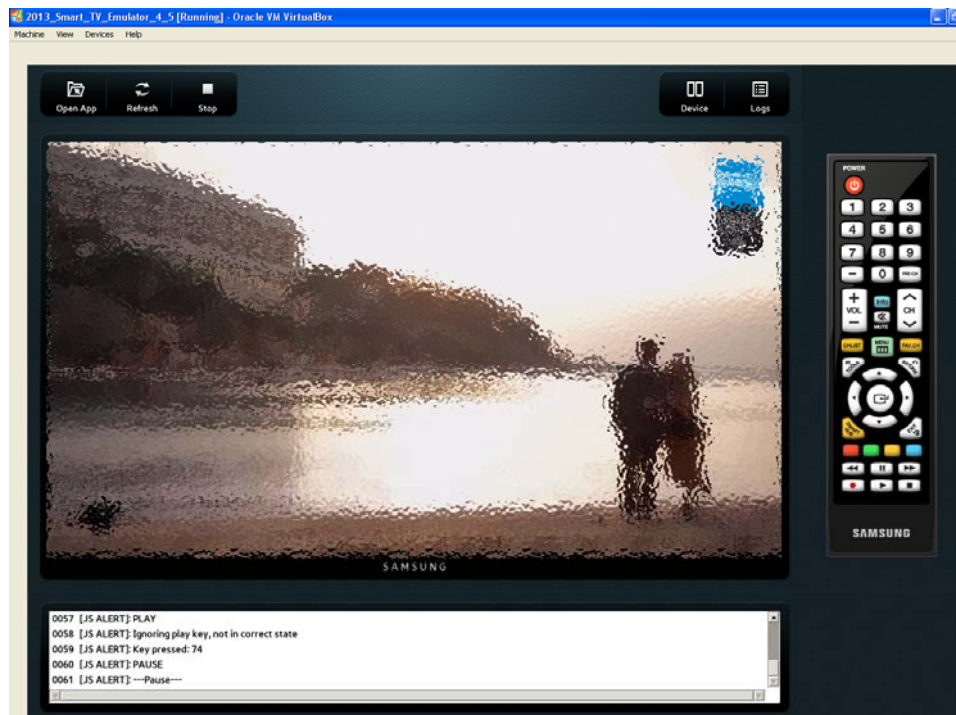
EIKONA 22

Μοντέλο 2013



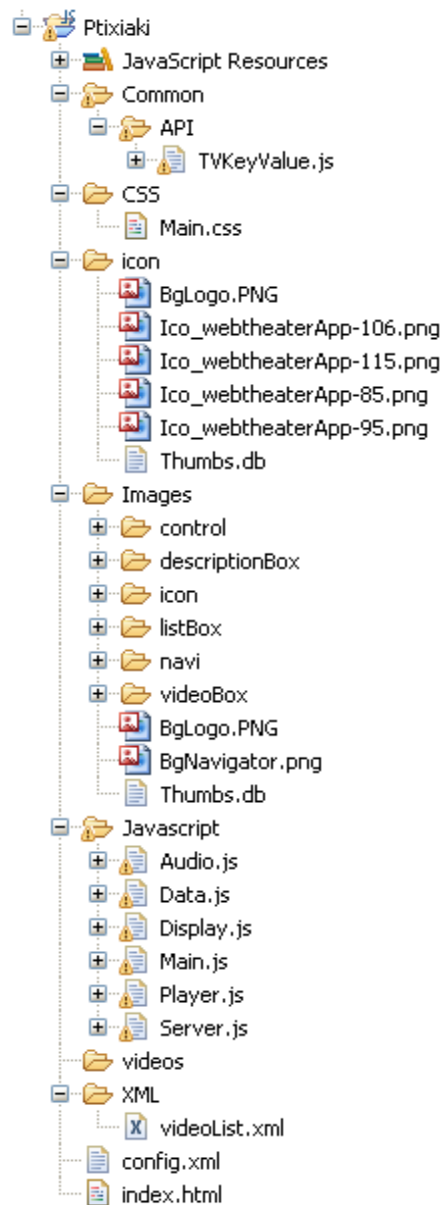
EIKONA 23

Μοντέλο 2013\_full screen



EIKONA 24

### 3.2.1 Διάρθρωση



EIKONA 25

Η εφαρμογή αποτελείται από τα αρχεία που έχουν εγκατασταθεί στην τηλεόραση και τον φάκελο με τα αρχεία video που βρίσκεται στον διακομιστή.

## Περιεχόμενα

- Common/API : **TVKeyValue** Object, Αυτό το αντικείμενο προσδιορίζει τα πλήκτρα του τηλεχειριστηρίου που έχουν πατηθεί από το χρήστη.
- CSS:περιέχει τα αρχεία css της εφαρμογής που είναι υπεύθυνα για την εμφάνιση
- Icon:περιέχει το αρχικό εικονίδιο της εφαρμογής
- Images:όλα τα αρχεία φωτογραφιών που χρησιμοποιεί η εφαρμογή
- Javascript:τα αρχεία javascript που χρησιμοποιεί η εφαρμογή
- config.xml:περιέχει τις πληροφορίες και κάποιες ρυθμίσεις που απαιτούνται για την εκτέλεση της εφαρμογής
- index.html:το αρχείο html που τρέχει η εφαρμογή

## Απομακρυσμένα στο server

- Videos:περιέχει τα αρχεία video που χρησιμοποιεί η εφαρμογή
- XML:περιέχει το αρχείο xml που τραβάει η εφαρμογή τις πληροφορίες για τα video

### 3.2.2 Μέρη εφαρμογής

- Control:υπεύθυνη κλάση για τον χειρισμό των ενεργειών του χρήστη και συντονίζει όλα τα κομμάτια της εφαρμογής
- TV : ελέγχει τα κομμάτια της τηλεόρασης
- Network: επικοινωνία με τον server
- Graphics: υπεύθυνο να εμφανίζει τις πληροφορίες και τα γραφικά στην index.html

| Πίνακας Κλάσεων      |
|----------------------|
| Control→Main.js      |
| TV→Player.js         |
| TV→Audio.js          |
| Network→Server.js    |
| Network→Data.js      |
| Graphics →Display.js |

EIKONA 26

### 3.2.3 Περιγραφή κώδικα

#### config.xml

Είναι το αρχείο με τις γενικές ρυθμίσεις της εφαρμογής

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.samsung.com/">
  <cpname itemtype="string"/>
  <cplogo itemtype="string"/>
  <cpauthjs itemtype="string"/>
  <ThumbIcon itemtype="string">icon/Ico_webtheaterApp-106.png</ThumbIcon>
  <BigThumbIcon itemtype="string">icon/Ico_webtheaterApp-115.png</BigThumbIcon>
  <ListIcon itemtype="string">icon/Ico_webtheaterApp-85.png</ListIcon>
  <BigListIcon itemtype="string">icon/Ico_webtheaterApp-95.png</BigListIcon>
  <category itemtype="string">Videos</category>
  <autoUpdate itemtype="boolean">n</autoUpdate>
  <ver itemtype="string">0.100</ver>
  <mgrver itemtype="string"/>
  <fullwidget itemtype="boolean">y</fullwidget>
  <movie itemtype="boolean">y</movie>
  <type itemtype="string">user</type>
  <srcctl itemtype="boolean">y</srcctl>
  <ticker itemtype="boolean">n</ticker>
  <childlock itemtype="boolean">n</childlock>
  <videomute itemtype="boolean">n</videomute>
  <audiomute itemtype="boolean">n</audiomute>
  <dcont itemtype="boolean">y</dcont>
  <widgetname itemtype="string">WebTheaterPlayer</widgetname>
  <description itemtype="string">Dimiourgia player gia to project webtheater.co
sta plaisia tis diplmatikis mou ergasias</description>
  <width itemtype="string">960</width>
  <height itemtype="string">540</height>
  <author itemtype="group">
    <name itemtype="string">Sofia Agrafioti</name>
    <email itemtype="string">-</email>
    <link itemtype="string">http://www.webtheater.co</link>
    <organization itemtype="string">-</organization>
  </author>
</widget>
```

EIKONA 27

| Xml Elenemt                                              | Περιγραφή                                                                                                |
|----------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <widget>                                                 | Υποδεικνύει ότι οι πληροφορίες είναι σχετικές με την εφαρμογή.                                           |
| <ThumbIcon>, <BigThumbIcon><br><ListIcon>, <BigListIcon> | Εικονίδια της εφαρμογής διαφόρων διαστάσεων<br><br>Διαστάσεις: 106 x 86, 115 x 95, 85 x 70, 95 x 78pixel |
| <category>                                               | Κατηγορία εφαρμογής                                                                                      |
| <autoUpdate>                                             | Αν θα συγχρονίζεται με αυτόματες ενημερώσεις με το site της Samsung                                      |
| <ver>                                                    | Έκδοση εφαρμογής                                                                                         |
| <mgrver>                                                 | Η έκδοση του application manager που χρειάζεται για να τρέχει η εφαρμογή                                 |

|                                     |                                                                                                                                                          |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>&lt;fullwidget&gt;</b>           | Επιτρέπει στην εφαρμογή να τρέχει και σε πλήρη οθόνη                                                                                                     |
| <b>&lt;srcctl&gt;</b>               | Αν είναι ενεργή η ρύθμιση αυτή γίνεται αυτόματη μετάβαση με την ολοκλήρωση της εφαρμογής από το εφαρμογή στο τρέχον κανάλι της τηλεόρασης και αντίστροφα |
| <b>&lt;childlock&gt;</b>            | Χρήση της λειτουργίας Κλείδωμα για παιδιά. Αυτή η λειτουργία δίνει τη δυνατότητα στο χρήστη να κλειδώσει την εφαρμογή.                                   |
| <b>&lt;audiomute&gt;</b>            | Ενεργοποιεί και απενεργοποιεί τον ήχο, Ρύθμιση στο "y" αν είναι ενεργοποιημένη η σίγαση κατά την είσοδο στην εφαρμογή.                                   |
| <b>&lt;videomute&gt;</b>            | Ενεργοποιεί και απενεργοποιεί το video. Ρύθμιση στο "y", τηλεοπτικές εκπομπές δεν είναι στην οθόνη κατά την είσοδό τους εφαρμογή.                        |
| <b>&lt;dcont&gt;</b>                | Ενεργοποίηση της λειτουργίας Disable dynamic contrast<br>Sets the Disable dynamic contrast function.                                                     |
| <b>&lt;movie&gt;</b>                | Η εφαρμογή μπορεί να υποστηρίξει video                                                                                                                   |
| <b>&lt;widgetname&gt;</b>           | Όνομα εφαρμογής                                                                                                                                          |
| <b>&lt;description&gt;</b>          | Περιγραφή εφαρμογής                                                                                                                                      |
| <b>&lt;width&gt; &lt;height&gt;</b> | Διαστάσεις εφαρμογής 960 * 540 pixels(προτεινόμενη )                                                                                                     |
| <b>&lt;author&gt;</b>               | Το όνομα του συγγραφέα                                                                                                                                   |
| <b>&lt;user&gt;</b>                 | Ενεργοποιεί τη δυνατότητα δοκιμής σε ένα πραγματική TV.                                                                                                  |
| <b>&lt;ticker&gt;</b>               | Ενεργοποίηση του ticker mode για autostart της εφαρμογής                                                                                                 |

EIKONA 28

## index.html

Η εφαρμογή, όταν είναι σε χρήση χρησιμοποιεί τον browser της τηλεόρασης, η index.html είναι το αρχείο με το layout της εφαρμογής και είναι αυτό που αλληλεπιδρά ο χρήστης, όταν είναι ενεργή η εφαρμογή.

Προκειμένου να μπορεί κάθε εφαρμογή να τρέξει στην τηλεόραση χρειάζεται να ενσωματωθούν κάποιες βιβλιοθήκες Common module.

- **TVKeyValue:** προσδιορίζει τα πλήκτρα του τηλεχειριστηρίου που έχουν πατηθεί από το χρήστη.
- **Widget:** διαθέτει τις λειτουργίες που απαιτούνται για την επιθυμητή λειτουργία της εφαρμογής. Είναι μια μέθοδος που ειδοποιεί το application manager(διαχειριστή εφαρμογής) για τη έναρξη της εφαρμογής.
- **Plugin:** Ο Διαχειριστής Εφαρμογών παρέχει τη δυνατότητα χρήσης wrapper class.

```

<!-- Common widget API -->
<script type='text/javascript' language='javascript' src='$MANAGER_WIDGET/Common/API/Widget.js'></script>
<script type='text/javascript' language='javascript' src='$MANAGER_WIDGET/Common/API/TVKeyValue.js'></script>
<script type='text/javascript' language='javascript' src='$MANAGER_WIDGET/Common/API/Plugin.js'></script>

```

EIKONA 29

Στην συνέχεια, αφού έχουν ενσωματωθεί οι απαραίτητες βιβλιοθήκες που χρειάζεται η έξυπνη τηλεόραση για την σωστή λειτουργία της ενσωματώνονται και τα javascript και css αρχεία της εφαρμογής.

```

<!-- Widget code -->
<script language="javascript" type="text/javascript" src="Javascript/Main.js"></script>
<script language="javascript" type="text/javascript" src="Javascript/Server.js"></script>
<script language="javascript" type="text/javascript" src="Javascript/Data.js"></script>
<script language="javascript" type="text/javascript" src="Javascript/Player.js"></script>
<script language="javascript" type="text/javascript" src="Javascript/Display.js"></script>
<script language="javascript" type="text/javascript" src="Javascript/Audio.js"></script>

<!-- Style sheets -->
<link rel="stylesheet" href="CSS/Main.css" type="text/css">

```

EIKONA 30

Η τελευταία ενσωμάτωση που χρειάζεται είναι προσθήκη κάποιων plugin :

- **player plugin:** για να μπορέσω να πάρω μια αναφορά του αντικειμένου player
- **Middlieware plugin:** για την σωστή λειτουργία video playback
- **Audio plugin:** για την αλλαγή της έντασης του ήχου

```

<!-- Plugins -->
<object id="pluginPlayer" border=0 classid="clsid:SAMSUNG-INFOLINK-PLAYER" ></object>
<object id="pluginAudio" border=0 classid="clsid:SAMSUNG-INFOLINK-AUDIO"></object>
<object id="pluginTVMW" border=0 classid="clsid:SAMSUNG-INFOLINK-TVMW"></object>

```

EIKONA 31

Τώρα η εφαρμογή έχει όλες τις απαραίτητες βιβλιοθήκες προκειμένου να μπορεί να λειτουργεί σωστά, στην συνέχεια ακολουθεί το body μέρος της index. Με χρήση anchor η εφαρμογή δέχεται τα κουμπιά που πατάει ο χρήστης. Στη συνέχεια ακολουθεί το layout της εφαρμογής.

Αναλυτικότερα το layout μπορούμε να το χωρίσουμε σε τέσσερις βασικές κλάσεις.

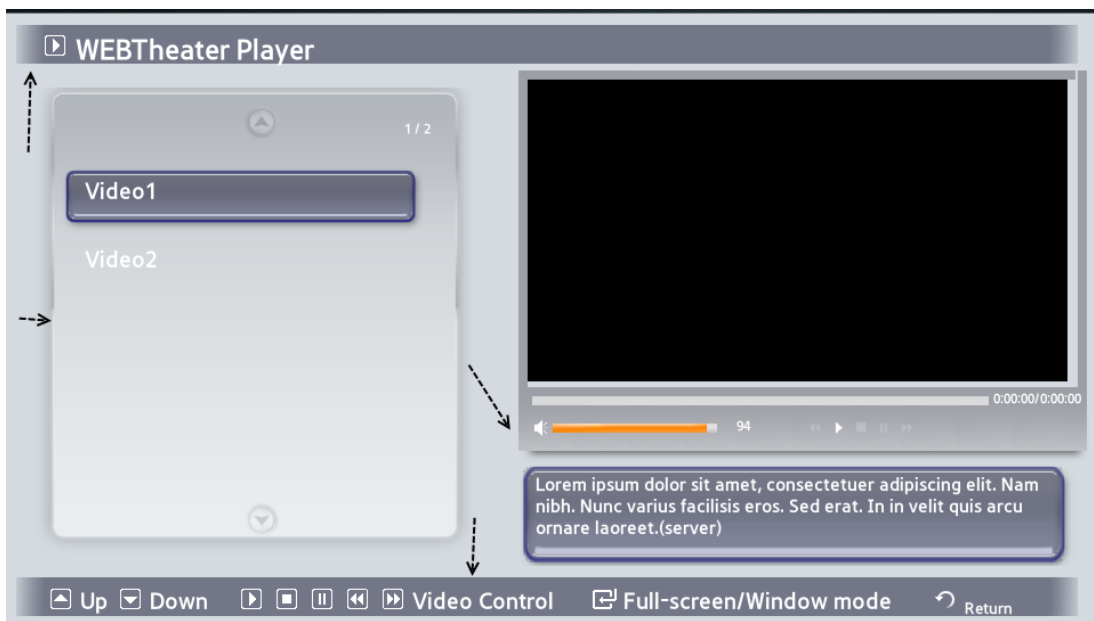
- Από τον τίτλο της εφαρμογής που εμφανίζεται ως μια μπάρα με τον τίτλο(`logo`)
- Το αριστερό κομμάτι την οθόνης που είναι η λίστα των παραστάσεων(`leftHalf`)
- Το δεξί κομμάτι που είναι ο `player` και μια ενδεικτική περιγραφή της παράστασης (`rightHalf`)
- Τέλος, μια βοηθητική μπάρα με επεξηγηματικά σύμβολα(`navi`).

```
<body onload="Main.onLoad();" onunload="Main.onUnload();" >
<!-- receive the key events -->
<a href='javascript:void(0);' id='anchor' onkeydown='Main.keyDown();'></a>

<!-- Layout -->
<div id="main">
    <div id="logo"> <span id="text">  </span> WEBTheater Player
</div>
<div id = "rightHalf">
    <div id="videoBox_top"></div>
    <div id="videoBox_left"></div>
    <div id="videoBox_right"></div>
    <div id="videoBox_bottom">
        <div id="time">
            <div id="progressBarBG">
                <div id="progressBar"></div>
            </div>
            <div id="timeInfo"></div>
        </div>
        <div id="volume">
            <div id="volumeIcon"></div>
            <div id="volumeBarBG">
                <div id="volumeBar"></div>
            </div>
            <div id="volumeInfo"></div>
        </div>
        <div id="videoControl">
            <div id="rewind"></div>
            <div id="play"></div>
            <div id="stop"></div>
            <div id="pause"></div>
            <div id="forward"></div>
        </div>
        <div id="status"></div>
    </div>
    <div id="description_top"></div>
    <div id="description_bottom"></div>
    <div id="description"></div>
</div>
<div id="images"></div>

<div id="leftHalf">
    <div id="videoList" class="style_videoList">
        <div id="video0"></div>
        <div id="video1"></div>
        <div id="video2"></div>
        <div id="video3"></div>
        <div id="video4"></div>

        <div id="videoCount"></div>
    </div>
    <div id="previous"></div>
    <div id="next"></div>
</div>
<div id="navi">
    <div id="help_navi">
        
        <a class="style_navi">Up</a> <a class="style_navi">Down</a> Video Control Full-screen/Window mode Return
    </div>
</div>
</body>
</html>
```

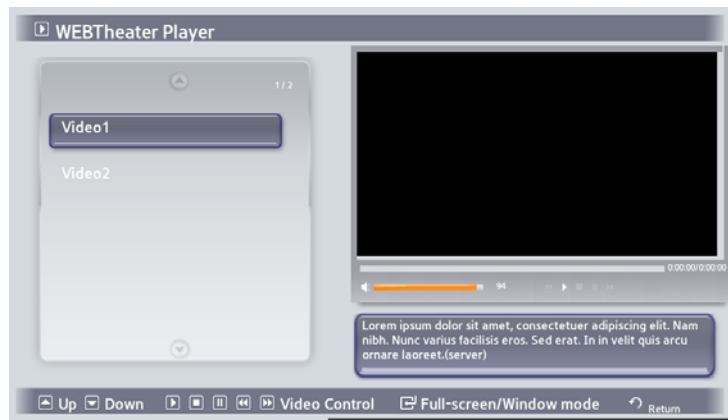


EIKONA 33

### Αρχικοποίηση εφαρμογής - main.js

Όταν ο χρήστης ανοίγει την εφαρμογή πραγματοποιείται αρχικοποίηση όλων των αντικειμένων. Στην επιτυχημένη είσοδο ο χρήστης θα είναι σε θέση να δει τη λίστα των διαθέσιμων video, την περιγραφή του πρώτου video και τον player του. Η Player.**init()** παίρνει αναφορά του αντικειμένου playerPlugin, Audio.**init()** του αντικειμένου του ήχου, Display.**init()** της display υπεύθυνης του time event με αρχική τιμή το μηδέν και τέλος Server.**init()** για να κάνει load τις πληροφορίες των video από το XML. Στις επόμενες ενότητες του κεφαλαίου αυτού θα αναλυθούν εκτενέστερα τα 4 βασικά αντικείμενα της εφαρμογής (Player, Audio, Display και Server). Τέλος, ενεργοποιείται και η function που διαχειρίζεται τα κουμπιά του τηλεχειριστηρίου.

Μια πιθανή εικόνα επιτυχημένης εισόδου, αν στον server υπήρχαν 2 video θα ήταν η ακόλουθη :



EIKONA 34

```

var widgetAPI = new Common.API.Widget();
var tvKey = new Common.API.TVKeyValue();

var Main =
(
    selectedVideo : 0,
    mode : 0,
    mute : 0,

    UP : 0,
    DOWN : 1,

    WINDOW : 0,
    FULLSCREEN : 1,

    NMUTE : 0,
    YMUTE : 1
);

Main.onLoad = function()
(
    if ( Player.init() && Audio.init() && Display.init() && Server.init() )
    {
        Display.setVolume( Audio.getVolume() );
        Display.setTime(0);

        Player.stopCallback = function()
        {
            /* Return to windowed mode when video is stopped
            (by choice or when it reaches the end) */
            Main.setWindowMode();
        }
    }

    // Start retrieving data from server
    Server.dataReceivedCallback = function()
    {
        /* Use video information when it has arrived */
        Display.setVideoList( Data.getVideoNames() );
        Main.updateCurrentVideo();
    }

    Server.fetchVideoList(); /* Request video information from server */

    // Enable key event processing
    this.enableKeys();

    widgetAPI.sendReadyEvent();
}
else
{
    alert("Failed to initialise");
}
var WIDGET_ID = curWidget.id;

alert("#WIDGET_ID# =" + WIDGET_ID);
);

```

EIKONA 35

Ο χρήστης με το που εισέρχεται στην εφαρμογή του έχει τη δυνατότητα να κάνει χρήση όλων των λειτουργιών που προσφέρει ένας τυπικός player μέσω του τηλεχειριστηρίου του, δηλαδή μπορεί:

- Να παίζει ένα video
- Να το σταματήσει
- Να πατήσει παύση
- Να το προχωρήσει μπροστά(αφού πρώτα έχει προηγηθεί η play)
- Να το πάει πίσω (αφού πρώτα έχει προηγηθεί η play)
- Να αυξήσει τον ήχο
- Να μειώσει τον ήχο
- Να ενεργοποιήσει την πλήρη οθόνη για να παρακολουθήσει μια παράσταση
- Να φύγει από την πλήρη οθόνη και να επανέλθει στην αρχική διάσταση



EIKONA 36

Για τη διευκόλυνση του χρήστη για το ποιες λειτουργίες μπορεί να χρησιμοποιήσει στο κάτω μέρος της εφαρμογής υπάρχει η βοηθητική μπάρα. Τα κουμπιά ενδέχεται να διαφέρουν από μοντέλο σε μοντέλο. Τα εικονίδια που έχουν χρησιμοποιηθεί στην μπάρα προέρχονται από το βασικό μοντέλο τηλεχειριστηρίου της Samsung.



EIKONA 37

Τα κουμπιά που απαιτούνται για τις δυνατότητες αυτές είναι τα ακόλουθα:

- Return
- Play
- Stop
- Pause
- Forward
- Back Forward
- Volume up
- Volume down
- Down
- Up
- Enter
- Mute

```

Main.enableKeys = function()
{
    document.getElementById("anchor").focus();
}

Main.keyDown = function()
{
    var keyCode = event.keyCode;
    alert("Key pressed: " + keyCode);

    switch(keyCode)
    {
        case tvKey.KEY_RETURN:
        case tvKey.KEY_PANEL_RETURN:
            alert("RETURN");
            Player.stopVideo();
            widgetAPI.sendReturnEvent();
            break;

        case tvKey.KEY_PLAY:
            alert("PLAY");

            this.handlePlayKey();
            break;

        case tvKey.KEY_STOP:
            alert("STOP");
            Player.stopVideo();
            break;

        case tvKey.KEY_PAUSE:
            alert("PAUSE");
            this.handlePauseKey();
            break;

        case tvKey.KEY_FF:
            alert("FF");
            if (Player.getState() != Player.PAUSED)
                Player.skipForwardVideo();
            break;

        case tvKey.KEY_RW:
            alert("RW");
            if (Player.getState() != Player.PAUSED)
                Player.skipBackwardVideo();
            break;

        case tvKey.KEY_VOL_UP:
        case tvKey.KEY_PANEL_VOL_UP:
            alert("VOL_UP");
            if (this.mute == 0)
                Audio.setRelativeVolume(0);
            break;

        case tvKey.KEY_VOL_DOWN:
        case tvKey.KEY_PANEL_VOL_DOWN:
            alert("VOL_DOWN");
            if (this.mute == 0)
                Audio.setRelativeVolume(1);
            break;

        case tvKey.KEY_DOWN:
            alert("DOWN");
            this.selectNextVideo(this.DOWN);
            break;

        case tvKey.KEY_UP:
            alert("UP");
            this.selectPreviousVideo(this.UP);
            break;

        case tvKey.KEY_ENTER:
        case tvKey.KEY_PANEL_ENTER:
            alert("ENTER");
            this.toggleNode();
            break;

        case tvKey.KEY_MUTE:
            alert("MUTE");
            this.muteNode();
            break;

        default:
            alert("Unhandled key");
            break;
    }
}

```

## EIKONA 38

Για τα κουμπιά του play και του pause χρειάζεται να προστεθούν δυο function που βοηθούν στη διαχείριση των πιθανών αλλαγών (stop, pause, play) που μπορούν να μεταβούν. Οι καταστάσεις αυτές θα μελετηθούν εκτενέστερα στην ενότητα του player.

```

Main.handlePlayKey = function()
{
    switch ( Player.getState() )
    {
        case Player.STOPPED:
            Player.playVideo();
            break;

        case Player.PAUSED:
            Player.resumeVideo();
            break;

        default:
            alert("Ignoring play key, not in correct state");
            break;
    }
}

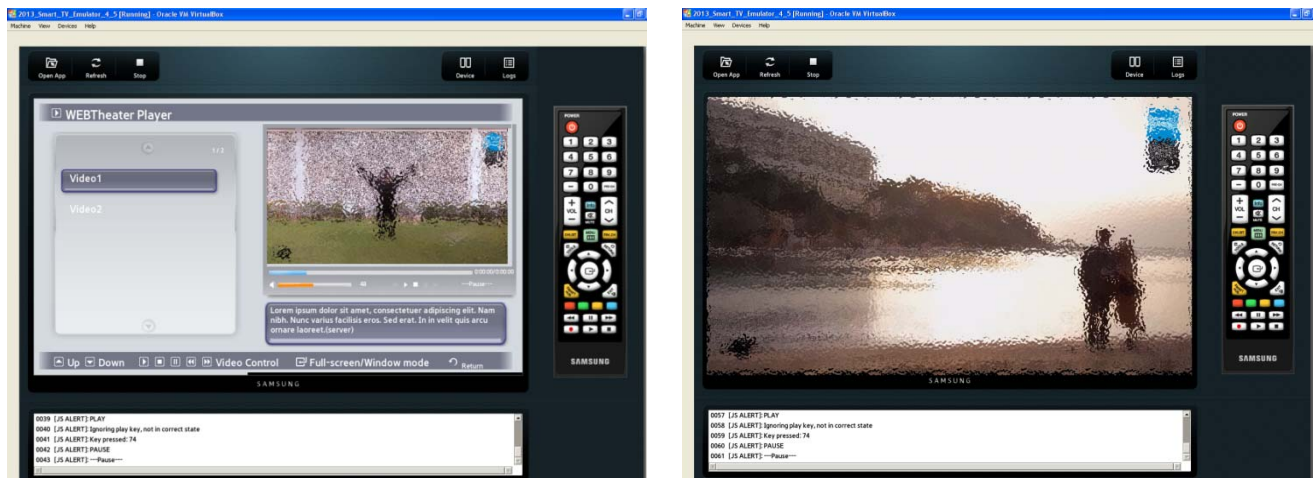
Main.handlePauseKey = function()
{
    switch ( Player.getState() )
    {
        case Player.PLAYING:
            Player.pauseVideo();
            break;

        default:
            alert("Ignoring pause key, not in correct state");
            break;
    }
}

```

EIKONA 39

Ο χρήστης έχει την δυνατότητα, αν το επιθυμεί να παρακολουθήσει την παράσταση σε πλήρη οθόνη μέσω του κουμπιού της πλήρους οθόνης. Η διάσταση της εφαρμογής από το configuration αρχείο έχει οριστεί στα 960x540.



EIKONA 40

```

Main.setFullScreenMode = function()
{
    if (this.mode != this.FULLSCREEN)
    {
        Display.hide();

        Player.setFullscreen();

        this.mode = this.FULLSCREEN;
    }
}

Main.setWindowMode = function()
{
    if (this.mode != this.WINDOW)
    {
        Display.show();

        Player.setWindow();

        this.mode = this.WINDOW;
    }
}

Main.toggleMode = function()
{
    if (Player.getState() == Player.PAUSED)
    {
        Player.resumeVideo();
    }
    switch (this.mode)
    {
        case this.WINDOW:
            this.setFullScreenMode();
            break;

        case this.FULLSCREEN:
            this.setWindowMode();
            break;

        default:
            alert("ERROR: unexpected mode in toggleMode");
            break;
    }
}

```

EIKONA 41

Επιπλέον ο χρήστης μπορεί:

- Να μετακινηθεί στη λίστα των video προς τα πάνω
- Να μετακινηθεί στη λίστα των video προς τα κάτω
- Να εξέλθει από την εφαρμογή και να επιστρέψει στο smart hub της Samsung με όλα τα application.

```

Main.updateCurrentVideo = function(move)
{
    Player.setVideoURL( Data.getVideoURL(this.selectedVideo) );

    Display.setVideoListPosition(this.selectedVideo);

    Display.setVideoListPosition(this.selectedVideo, move);

    Display.setDescription( Data.getVideoDescription(this.selectedVideo));
}

```

```

Main.selectNextVideo = function(down)

```

```

{
    Player.stopVideo();

    this.selectedVideo = (this.selectedVideo + 1) % Data.getVideoCount();

    this.updateCurrentVideo(down);
}

Main.selectPreviousVideo = function(up)
{
    Player.stopVideo();

    if (--this.selectedVideo < 0)
    {
        this.selectedVideo += Data.getVideoCount();
    }

    this.updateCurrentVideo(up);
}

```

EIKONA 42

Η καρτέλα των video έχει χωρητικότητα μέχρι 5 τίτλους παραστάσεων, αν όμως το xml αρχείο περιέχει παραπάνω από 5 παραστάσεις τα άλλα video είναι προσπελάσιμα μέσω της λειτουργίας up /down του τηλεχειριστηρίου. Ο συνολικός αριθμός των video είναι ορατός στο πάνω μέρος της λίστας.



EIKONA 43

```

<div id="videoList" class="style_videoList">
    <div id="video0"></div>
    <div id="video1"></div>
    <div id="video2"></div>
    <div id="video3"></div>
    <div id="video4"></div>
    <div id="videoCount"></div>
</div>

```

EIKONA 44

## Επικοινωνία με server

Όπως αναφέρθηκε στο κεφάλαιο της αρχικοποίησης, όταν ανοίγει ο χρήστης την εφαρμογή αρχικοποιούνται όλα τα plugin της εφαρμογής και πραγματοποιείται επικοινωνία με τον server προκειμένου να εμφανιστεί η λίστα των παραστάσεων μαζί με τη σύντομη περιγραφή της. Για δοκιμαστικούς λόγους στα πλαίσια της διπλωματικής χρησιμοποιήθηκε ένας server της σχολής.

```
url : "http://galaxy.hua.gr/~itp12201/XML/videoList.xml"
```

Για να επικοινωνήσει η εφαρμογή με τον server και να ανακτήσει τα data από το xml γίνεται χρήση των XmlHttpRequest(XHR).

Στο αρχικό request που πραγματοποιείται στον server, αν επιστραφεί σαν response η **κατάσταση 4** (request finished and response is ready) και **το status 200(OK)** τότε καλείται η function createVideoList που δημιουργεί την λίστα.

```
/* function pou fernei tin lista */
Server.fetchVideoList = function()
{
    if (this.XHRObj == null)
    {
        this.XHRObj = new XMLHttpRequest();
    }

    if (this.XHRObj)
    {
        this.XHRObj.onreadystatechange = function()
        {
            if (Server.XHRObj.readyState == 4 && Server.XHRObj.status == 200 )
            {
                Server.createVideoList();
            }
        }

        this.XHRObj.open("GET", this.url, true);
        this.XHRObj.send(null);
    }
    else
    {
        alert("Failed to create XHR");
    }
}
```

EIKONA 45

Η `createVideoList` δημιουργεί τη λίστα με τις πληροφορίες του αντικειμένου `data`, το αντικείμενο περιλαμβάνει 3 πληροφορίες :

- τον τίτλο του video
- σύντομη περιγραφή
- και το url του video

```
Server.createVideoList = function()
{
    if (this.XHRObj.status != 200)
    {
        Display.status("XML Server Error " + this.XHRObj.status);
    }
    else
    {
        var xmlElement = this.XHRObj.responseXML.documentElement;

        if (!xmlElement)
        {
            alert("Failed to get valid XML");
        }
        else
        {
            // Get all "item" elements
            var items = xmlElement.getElementsByTagName("item");

            var videoNames = [ ];
            var videoURLs = [ ];
            var videoDescriptions = [ ];

            for (var index = 0; index < items.length; index++)
            {
                var titleElement = items[index].getElementsByTagName("title")[0];
                var descriptionElement = items[index].getElementsByTagName("description")[0];
                var linkElement = items[index].getElementsByTagName("link")[0];

                if (titleElement && descriptionElement && linkElement)
                {
                    videoNames[index] = titleElement.firstChild.data;
                    videoURLs[index] = linkElement.firstChild.data;
                    videoDescriptions[index] = descriptionElement.firstChild.data;
                }
            }

            Data.setVideoNames(videoNames);
            Data.setVideoURLs(videoURLs);
            Data.setVideoDescriptions(videoDescriptions);

            if (this.dataReceivedCallback)
            {
                this.dataReceivedCallback(); /* enimerosi oti ola ta data exoun lifthei */
            }
        }
    }
}
```

EIKONA 46

Τέλος, στην επιτυχημένη επικοινωνία και τη λήψη των πληροφοριών το XHR αντικείμενο αποδεσμεύεται για να μη γίνεται άσκοπη χρήση μνήμης.

```

Server.init = function()
{
    var success = true;

    if (this.XHRobj)
    {
        this.XHRobj.destroy(); // epituximeni epikoinonia apodesmeusi tou
        antikeimenoy
        this.XHRobj = null;
    }

    return success;
}

```

EIKONA 47

## Setter Getters methods

Στο Data.js βρίσκονται οι setters and getters μέθοδοι του αντικειμένου Data με τις 3 ιδιότητες

- τον τίτλο του video,
- σύντομη περιγραφή
- και το url του video.

```

var Data =
{
    videoNames : [ ],
    videoURLs : [ ],
    videoDescriptions : [ ],
}

Data.setVideoNames = function(list)
{
    this.videoNames = list;
}

Data.setVideoURLs = function(list)
{
    this.videoURLs = list;
    alert("video urls:" +this.videoURLs + " ");
}

Data.setVideoDescriptions = function(list)
{
    this.videoDescriptions = list;
    alert("Description:"+this.videoDescriptions);
}

Data.getVideoURL = function(index)
{
    var url = this.videoURLs[index];

    if (url)    // Check for undefined entry
    {
        alert("URL is: "+url);
        return url;
    }
    else
    {
        return null;
    }
}

Data.getVideoCount = function()
{
    return this.videoURLs.length;
}

Data.getVideoNames = function()
{
    return this.videoNames;
}

Data.getVideoDescription = function(index)
{
    var description = this.videoDescriptions[index];

    if (description)    // Check for undefined entry
    {
        return description;
    }
    else
    {
        return "No description";
    }
}

```

EIKONA 48

## XML/videoList.xml

Από το αρχείο xml αυτό διαβάζονται:

- ο τίτλος,
- το url του video
- και μια μικρή περιγραφή της κάθε παράστασης.

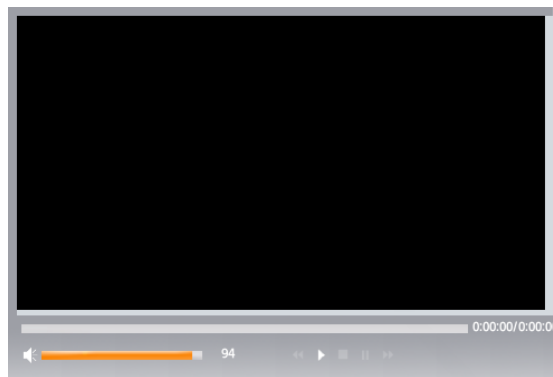
To videoList.xml βρίσκεται στον server

```
<item>
  <title>Video1</title>
  <link>http://galaxy.hua.gr/~itp12201/videos/movie.mp4</link>
  <description>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Nam nibh. Nunc varius facilisis eros.
    Sed erat. In in velit quis arcu ornare laoreet.(server)
  </description>
</item>
```

EIKONA 49

## Λειτουργίες Video Playback Buffer

Στην ενότητα αυτή θα περιγραφεί η λειτουργία του Player. Η Samsung υποστηρίζει τη λειτουργία Playback buffer που χρησιμοποιεί και η εφαρμογή για να παίζει τα video. Η playback επιτρέπει στον χρήστη να αναπαράγει το video που θέλει, αφού πρώτα το κρατήσει προσωρινά στη μνήμη της συσκευής του.



EIKONA 50

Οι αρχικές τιμές των ιδιοτήτων του αντικειμένου Player

```
var Player =
{
  plugin : null,
  state : -1,
  skipState : -1,
  stopCallback : null, /* Callback function */
  originalSource : null,
```

```

/* koumplia player */
STOPPED : 0,
PLAYING : 1,
PAUSED  : 2,
FORWARD : 3,
REWIND  : 4
}

```

EIKONA 51

Το playerPlugin θα ξεκινήσει με τις ακόλουθες προδιαγραφές:

- Θα είναι στην κατάσταση stop , ενεργό κουμπί στον player μόνο το play
- Η διάσταση της αρχικής οθόνης του video είναι 472x270
- Θα έχει μηδενισμένη τη τρέχουσα κατάσταση της χρόνου του video
- Θα έχει μηδενισμένη την συνολική διάρκεια του

Το PlayerPlugin διαβάζει και τις ακόλουθες πληροφορίες:

- Την τρέχουσα κατάσταση της χρόνου του video
- Τη συνολική διάρκεια του video
- Πότε ξεκινάει η διαδικασία του buffering των video για το playback
- Την πρόοδο του buffering
- Την ολοκλήρωση του buffering

```

Player.init = function()
{
    var success = true;
    alert("player success : " + success);
    this.state = this.STOPPED;

    this.plugin = document.getElementById("pluginPlayer");
    alert("plugin"+this.plugin);
    if (!this.plugin)
    {
        alert("success this.plugin : " + success);
        success = false;
    }

    alert("success : " + success);

    this.setWindow();

    alert("success : " + success);

    this.plugin.OnCurrentPlayTime = 'Player.setCurTime';
    this.plugin.OnStreamInfoReady = 'Player.setTotalTime';
    this.plugin.OnBufferingStart = 'Player.onBufferingStart';
    this.plugin.OnBufferingProgress = 'Player.onBufferingProgress';
    this.plugin.OnBufferingComplete = 'Player.onBufferingComplete';

    alert("success : " + success);
    return success;
}

```

EIKONA 52

Αποδέσμευση του player για εξοικονόμηση μνήμης

```

Player.deinit = function()
{
    alert("Player deinit !!! ");

    if (this.plugin)
    {
        this.plugin.Stop();
    }
}

```

EIKONA 53

Ορισμός των διαστάσεων του αρχικού παράθυρου και της πλήρους οθόνης.

```

Player.setWindow = function()
{
    alert("setting window");
    this.plugin.SetDisplayArea(458, 58, 472, 270);
}
Player.setFullscreen = function()
{
    this.plugin.SetDisplayArea(0, 0, 960, 540);
}

```

EIKONA 54

Όλα τα κουμπιά του player δεν είναι ενεργά σε όλες τις περιπτώσεις αλλά εξαρτάται σε ποια κατάσταση έχει αφήσει τον player του ο χρήστης.

Για λόγους ευχρηστίας της εφαρμογής το κουμπί που έχει πατηθεί και είναι ενεργό παρουσιάζεται πιο αχνό από τα άλλα για να θυμάται ποιο έχει επιλέξει, επιπλέον εμφανίζεται στην οθόνη του και με λεκτικό σε ποια κατάσταση βρίσκεται.

Οι περιπτώσεις που μπορούν να υπάρξουν είναι:

- Από την κατάσταση play μπορεί να μεταβεί στην pause, stop.
- Από την pause στην resume, stop
- Από την stop στην play

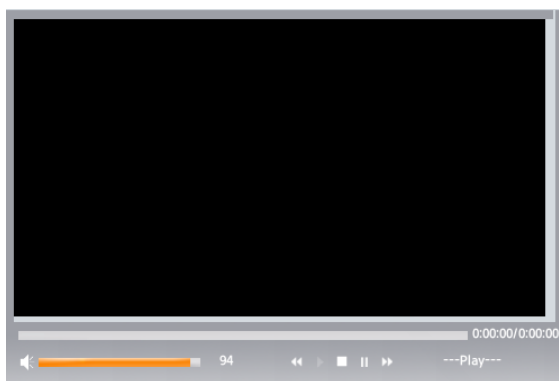
Οι λειτουργίες forward και back forward μπορούν να λειτουργήσουν, μόνο αν έχει ξεκινήσει η διαδικασία του buffering. Πριν ξεκινήσει να παίζει το video δεν μπορεί ο χρήστης να το προχωρήσει, όποτε πρέπει υποχρεωτικά να έχει πατηθεί πρώτα το play και μόνο από την play.

Στην συνέχεια ακολουθεί ένα σχηματικό παράδειγμα με τις καταστάσεις.

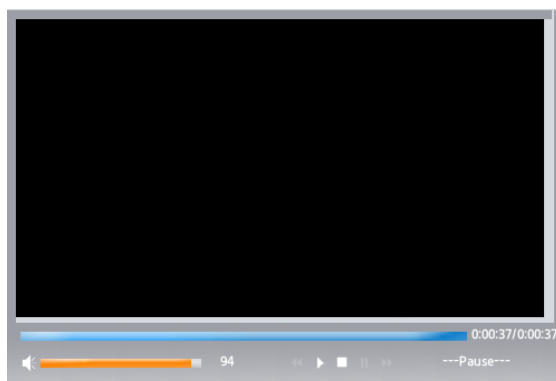
Το παρακάτω σχήμα δείχνει :

- Την τρέχουσα κατάσταση του Player
- Τα ενεργά κουμπιά κάθε κατάστασης,
- Την πρόοδο του χρόνου του video αριθμητικά και γραφικά
- Τέλος, την ένταση του ήχου αριθμητικά και γραφικά

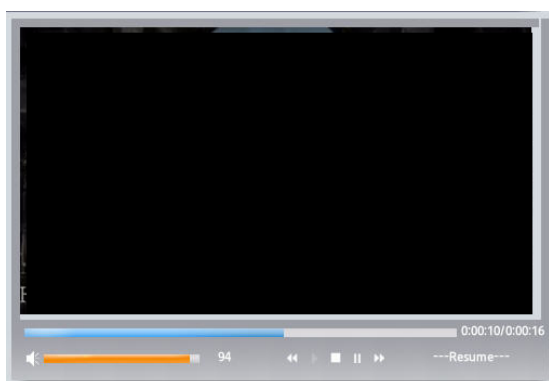
Play



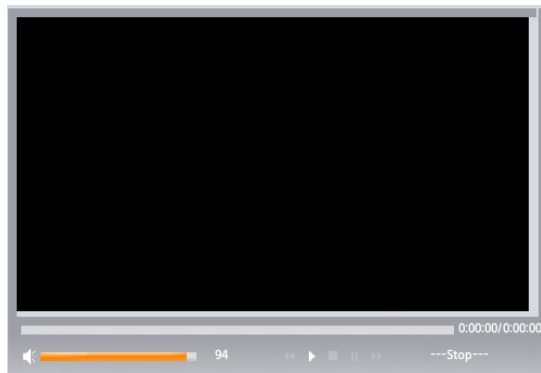
Pause



Resume



Stop



ΕΙΚΟΝΑ 55 Μετάβαση από την κατάσταση play → pause → στην play – resume → stop

```

Player.setVideoURL = function(url)
{
    this.url = url;
    alert("URL = " + this.url);
}

Player.playVideo = function()
{
    if (this.url == null)
    {
        alert("No videos to play");
    }
    else
    {
        this.state = this.PLAYING;

        /* css value 0.2 for not press, 1.0 for press*/
        document.getElementById("play").style.opacity = '0.2';
        document.getElementById("stop").style.opacity = '1.0';
        document.getElementById("pause").style.opacity = '1.0';
        document.getElementById("forward").style.opacity = '1.0';
        document.getElementById("rewind").style.opacity = '1.0';
        Display.status("---Play---");
        this.plugin.Play(this.url);
    }
}

Player.pauseVideo = function()
{
    this.state = this.PAUSED;
    document.getElementById("play").style.opacity = '1.0';
    document.getElementById("stop").style.opacity = '1.0';
    document.getElementById("pause").style.opacity = '0.2';
    document.getElementById("forward").style.opacity = '0.2';
    document.getElementById("rewind").style.opacity = '0.2';
    Display.status("---Pause---");
    this.plugin.Pause();
}

Player.stopVideo = function()
{
    if (this.state != this.STOPPED)
    {
        this.state = this.STOPPED;
        document.getElementById("play").style.opacity = '1.0';
        document.getElementById("stop").style.opacity = '0.2';
        document.getElementById("pause").style.opacity = '0.2';
        document.getElementById("forward").style.opacity = '0.2';
        document.getElementById("rewind").style.opacity = '0.2';
        Display.status("---Stop---");
        this.plugin.Stop();
        Display.setTime(0);

        if (this.stopCallback)
        {
            this.stopCallback();
        }
    }
    else
    {
        alert("Ignoring stop request, not in correct state");
    }
}

Player.resumeVideo = function()
{
    this.state = this.PLAYING;
    document.getElementById("play").style.opacity = '0.2';
    document.getElementById("stop").style.opacity = '1.0';
    document.getElementById("pause").style.opacity = '1.0';
    document.getElementById("forward").style.opacity = '1.0';
    document.getElementById("rewind").style.opacity = '1.0';
    Display.status("---Resume---");
    this.plugin.Resume();
}

Player.skipForwardVideo = function()
{
    this.skipState = this.FORWARD;
    this.plugin.JumpForward(3);
}

Player.skipBackwardVideo = function()
{
    this.skipState = this.REWIND;
    this.plugin.JumpBackward(3);
}

Player.getState = function()
{
    return this.state;
}

// Oj leiturqies forward kai backforward einai energez mono otan exeì ksekinisei to buffering

Player.onBufferingStart = function()
{
    Display.status("Buffering...");
    switch(this.skipState)
    {
        case this.FORWARD:
            document.getElementById("forward").style.opacity = '0.2';
            break;

        case this.REWIND:
            document.getElementById("rewind").style.opacity = '0.2';
            break;
    }
}

```

Στην αρχή της ενότητας αυτής αναφέρθηκε ότι απαιτείται το Video να αποθηκευτεί στη μνήμη πριν ξεκινήσει να παίζει, οπότε πραγματοποιούνται κάποιοι έλεγχοι προόδου της διαδικασίας αυτής. Πέρα από την κατάσταση του buffering παρακολουθείται και η σωστή λειτουργία του δικτύου και υπολογισμός του χρόνου του εκάστοτε Video.

```

Player.onBufferingProgress = function(percent)
{
    var fileUrl = this.url;

    alert(fileUrl);
    Display.status("Buffering:" + percent + "%");
}

Player.onBufferingComplete = function()
{
    Display.status("Play");
    switch(this.skipState)
    {
        case this.FORWARD:
            document.getElementById("forward").style.opacity = '1.0';
            break;

        case this.REWIND:
            document.getElementById("rewind").style.opacity = '1.0';
            break;
    }
}

stopPlayer = function()
{
    Player.stopVideo();
}

Player.setCurTime = function(time)
{
    Display.setTime(time);
}

Player.setTotalTime = function()
{
    Display.setTotalTime(Player.plugin.GetDuration());
}

/* Διαφοροι ελεχοι για τον server το diktio tou bandwidth */
onServerError = function()
{
    Display.status("Server Error!");
}

OnNetworkDisconnected = function()
{
    Display.status("Network Error!");
}

getBandwidth = function(bandwidth) { alert("getBandwidth " + bandwidth); }

onDecoderReady = function() { alert("onDecoderReady"); }

/* ελεχνος apodosois
    Unsupported container
    Unsupported video codec
    Unsupported audio codec
    Unsupported video resolution
*/
onRenderError = function() { alert("onRenderError"); }

/* Buffering*/
setTottalBuffer = function(buffer) { alert("setTottalBuffer " + buffer); }

setCurBuffer = function(buffer) { alert("setCurBuffer " + buffer); }

```

## Διαχείριση - Εμφάνιση Λειτουργιών ήχου

Προκειμένου η εφαρμογή να μπορεί να διαχειριστεί τις λειτουργίες του ήχου στην function που ελέγχει τα κουμπιά που πατάει ο χρήστης προστίθενται και οι λειτουργίες του ήχου volume up, down, mute.

Στην main.js έχει οριστεί ότι το κουμπί UP θα έχει την τιμή 0 και το κουμπί DOWN την τιμή 1.

```
case tvKey.KEY_PANEL_VOL_UP:
    alert("VOL_UP");
    if(this.mute == 0)
        Audio.setRelativeVolume(0);
    break;

case tvKey.KEY_VOL_DOWN:
case tvKey.KEY_PANEL_VOL_DOWN:
    alert("VOL_DOWN");
    if(this.mute == 0)
        Audio.setRelativeVolume(1);
    break;
case tvKey.KEY_MUTE:
    alert("MUTE");
    this.muteMode();
    break;
```

EIKONA 58

Το αρχείο audio.js είναι υπεύθυνο για την αλλαγή του ήχου, η μέγιστη τιμή που μπορεί να φτάσει είναι το 100 και το ελάχιστο είναι το 0. Κάθε φορά που πατάει ο χρήστης UP ή DOWN η ένταση του ήχου αλλάζει κατά μια μονάδα .

```
var Audio =
{
    plugin : null
},

Audio.init = function()
{
    var success = true;

    this.plugin = document.getElementById("pluginAudio");

    if (!this.plugin)
    {
        success = false;
    }

    return success;
},

Audio.setRelativeVolume = function(vol)
{
    this.plugin.SetVolumeWithKey(vol);
    Display.setVolume( this.getVolume() );
},

Audio.getVolume = function()
{
    alert("Volume : " + this.plugin.GetVolume());
    return this.plugin.GetVolume();
},
```

EIKONA 59

## Λειτουργία Mute

Η λειτουργία mute λειτουργεί σε 2 καταστάσεις on/off(toggle mode), οπότε δεν ελέγχεται από την audio.js αλλά έχουν προστεθεί στην main 2 function που διαχειρίζονται τις 2 καταστάσεις on/off.

```
Main.setMuteMode = function()
{
    if (this.mute != this.YMUTE)
    {
        var volumeElement = document.getElementById("volumeInfo");
        //Audio.plugin.SetSystemMute(true);
        Audio.plugin.SetUserMute(true);
        document.getElementById("volumeBar").style.backgroundColor = "url(Images/videoBox/muteBar.png)";
        document.getElementById("volumeIcon").style.backgroundColor = "url(Images/videoBox/mute.png)";
        widgetAPI.putInnerHTML(volumeElement, "MUTE");
        this.mute = this.YMUTE;
    }
}

Main.noMuteMode = function()
{
    if (this.mute != this.NMUTE)
    {
        Audio.plugin.SetUserMute(false);
        document.getElementById("volumeBar").style.backgroundColor = "url(Images/videoBox/volumeBar.png)";
        document.getElementById("volumeIcon").style.backgroundColor = "url(Images/videoBox/volume.png)";
        Display.setVolume( Audio.getVolume() );
        this.mute = this.NMUTE;
    }
}

Main.muteMode = function()
{
    switch (this.mute)
    {
        case this.NMUTE:
            this.setMuteMode();
            break;

        case this.YMUTE:
            this.noMuteMode();
            break;

        default:
            alert("ERROR: unexpected mode in muteMode");
            break;
    }
}
```

EIKONA 60

Στο layout της εφαρμογής οι αλλαγές του ήχου εμφανίζονται με μια progress bar με την τρέχουσα τιμή του ήχου. Στην index βρίσκεται εσωτερικά του div *rightHalf*.

```

<div id="volume">
  <div id="volumeIcon"></div>
  <div id="volumeBarBG">
    <div id="volumeBar"></div>
  </div>
  <div id="volumeInfo"></div>
</div>

```

## EIKONA 61

```

#volume
{
  position: absolute;
  left: 8px; top: 24px;
  width: 472px; height: 24px;
}

#volumeIcon
{
  position: absolute;
  left: 4px; top: 4px;
  width: 17px; height: 16px;
  background-image: url("../Images/videoBox/volume.png");
}

#volumeBarBG
{
  position: absolute;
  left: 22px; top: 8px;
  width: 144px; height: 8px;
  background-image: url("../Images/videoBox/volumeBarBG.png");
}

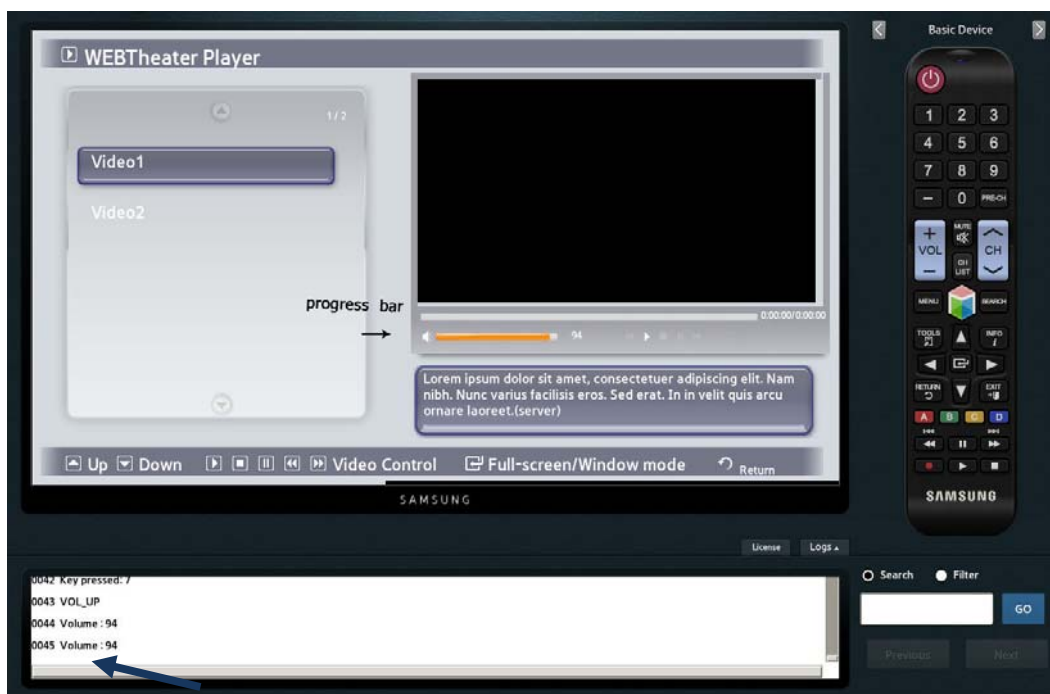
#volumeBar
{
  position: absolute;
  left: 0px; top: 0px;
  width: 0%; height: 8px;
  background-image: url("../Images/videoBox/volumeBar.png");
}

#volumeInfo
{
  position: absolute;
  left: 168px; top: 4px;
  width: 32px; height: 16px;

  text-align: left;
  font-size: 12px;
}

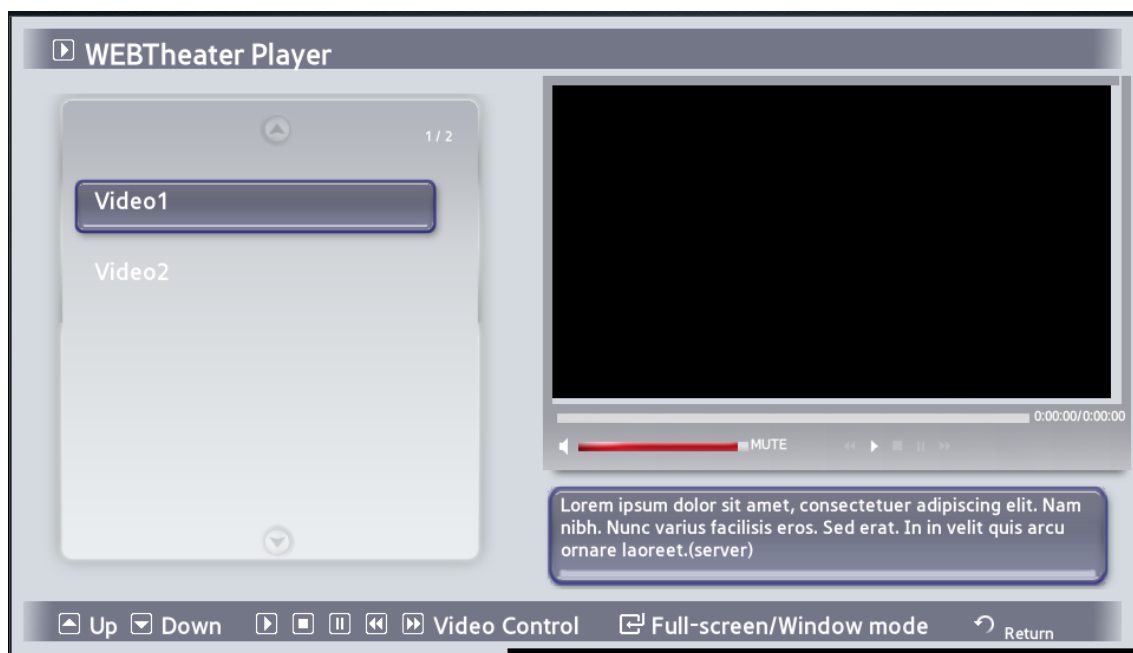
```

## EIKONA 62



EIKONA 63

Ενεργή η κατάσταση Mute



EIKONA 64

Τέλος, για να εμφανιστούν οι αυξομειώσεις του ήχου στην progress bar προστίθεται ο παρακάτω κώδικας:

```
Display.setVolume = function(level)
{
    document.getElementById("volumeBar").style.width = level + "%";

    var volumeElement = document.getElementById("volumeInfo");

    widgetAPI.putInnerHTML(volumeElement, "       " +
Audio.getVolume());
}
```

EIKONA 65



EIKONA 66

## Εμφάνιση όλων των αντικειμένων στην εφαρμογή

Στην ενότητα αυτή θα περιγραφεί το τέταρτο και τελευταίο αντικείμενο, η Display που είναι υπεύθυνη να εμφανίζει τα προηγούμενα αντικείμενα στην εφαρμογή. Όπως και προηγουμένως ορίζονται οι αρχικές τιμές.

Στην ενότητα του player αναφέρθηκε ότι για διευκόλυνση του χρήστη εμφανίζουμε και με λεκτικό την κατάσταση του Player. Όταν ο χρήστης ανοίγει την εφαρμογή η κατάσταση είναι κενή και δεν εμφανίζει τίποτα **statusDiv : null**



EIKONA 67

```

var Display =
{
    statusDiv : null,      /* css class status */
    FIRSTIDX : 0,         /* first video*/
    LASTIDX : 4,          /* last video */
    currentWindow : 0,

    SELECTOR : 0,         /* koumpia up /down */
    LIST : 1,            /* arithmos 1ou video */

    videoList : new Array()
}

Display.init = function()
{
    var success = true;

    this.statusDiv = document.getElementById("status");

    if (!this.statusDiv)
    {
        success = false;
    }

    return success;
}

Display.status = function(status)
{
    alert(status);
    widgetAPI.putInnerHTML(this.statusDiv, status);
}

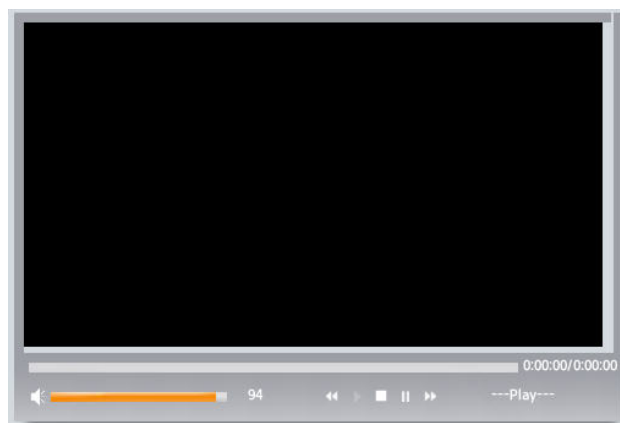
```

EIKONA 68

Ακολουθεί ο υπολογισμός της διάρκειας του video. Ο χρόνος εμφανίζεται στον χρήστη με 2 τρόπους, ο πρώτος είναι με τη μορφή μιας progress bar και ο δεύτερος με αριθμούς της μορφής ώρα: λεπτά: δευτερόλεπτα(0:00:00/0:00:00). Ο πρώτος αριθμός αντιπροσωπεύει την τρέχουσα χρονική τιμή του video, ενώ η δεύτερη είναι η συνολική διάρκεια

Όταν το video ξεκινήσει να παίζει έχει κάνει buffer, δηλαδή τότε εμφανίζεται γραφικά με μια bar σε τι στάδιο είναι, καθώς εμφανίζεται ο τωρινός χρόνος σε σχέση με την συνολική διάρκεια.

Διαδικασία buffer – play status



Πρόοδος video



Ολοκλήρωση video



EIKONA 69

```

/* ypologismos diarkeias */
Display.setTotalTime = function(total)
{
    this.totalTime = total;
}
Display.setTime = function(time)
{
    var timePercent = (100 * time) / this.totalTime;
    var timeElement = document.getElementById("timeInfo");
    var timeHTML = "";
    var timeHour = 0; var timeMinute = 0; var timeSecond = 0;
    var totalTimeHour = 0; var totalTimeMinute = 0; var totalTimeSecond = 0;

    document.getElementById("progressBar").style.width = timePercent + "%";

    if(Player.state == Player.PLAYING)
    {
        totalTimeHour = Math.floor(this.totalTime/3600000);
        timeHour = Math.floor(time/3600000);

        totalTimeMinute = Math.floor((this.totalTime%3600000)/60000);
        timeMinute = Math.floor((time%3600000)/60000);

        totalTimeSecond = Math.floor((this.totalTime%60000)/1000);
        timeSecond = Math.floor((time%60000)/1000);

        timeHTML = timeHour + ":";

        if(timeMinute == 0)
            timeHTML += "00:";
        else if(timeMinute <10)
            timeHTML += "0" + timeMinute + ":";
        else
            timeHTML += timeMinute + ":";

        if(timeSecond == 0)
            timeHTML += "00/";
        else if(timeSecond <10)
            timeHTML += "0" + timeSecond + "/";
        else
            timeHTML += timeSecond + "/";

        timeHTML += totalTimeHour + ":";

        if(totalTimeMinute == 0)
            timeHTML += "00:";
        else if(totalTimeMinute <10)
            timeHTML += "0" + totalTimeMinute;
        else
            timeHTML += totalTimeMinute;

        if(totalTimeSecond == 0)
            timeHTML += "00";
        else if(totalTimeSecond <10)
            timeHTML += "0" + totalTimeSecond;
        else
            timeHTML += totalTimeSecond;
    }
    else
        timeHTML = "0:00:00/0:00:00";

    widgetAPI.putInnerHTML(timeElement, timeHTML);
}

```

EIKONA 70

Ίδιος τρόπος εμφάνισης ακολουθείται και για τον ήχο, υπάρχει δηλαδή μια progress bar αλλά και αριθμητική τιμή του επιπέδου του ήχου.



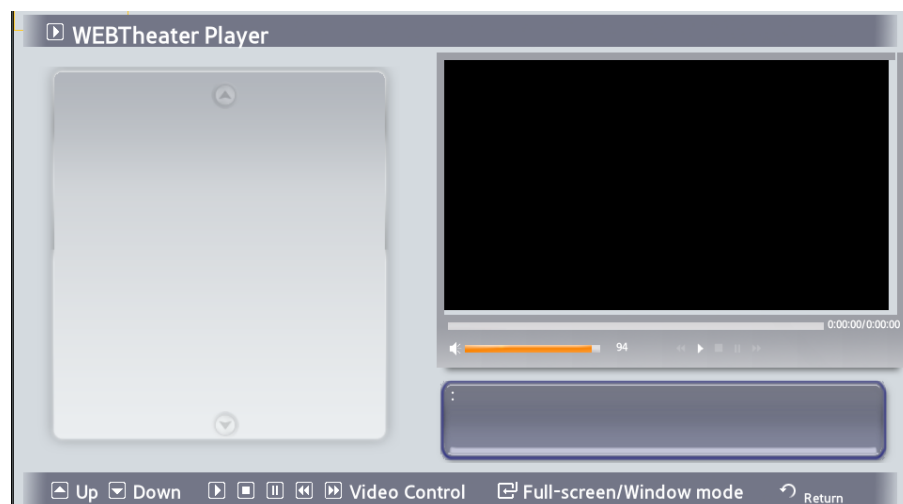
EIKONA 71

[illegible]

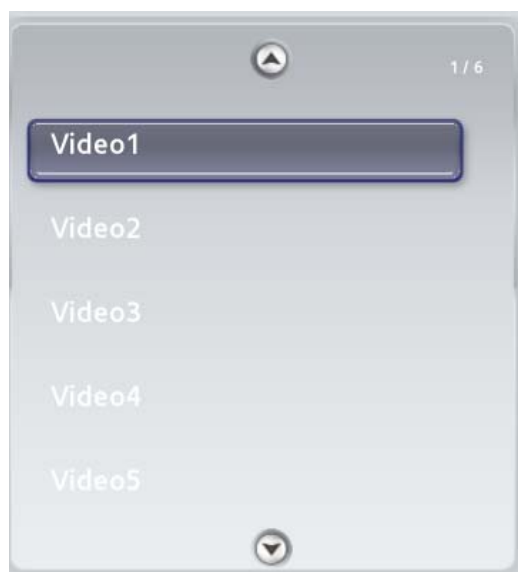
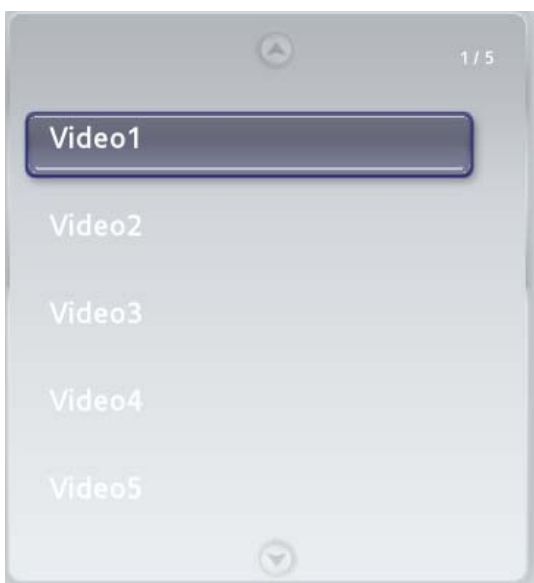
EIKONA 72

Στη συνέχεια ακολουθεί η εμφάνιση της αριστερής στήλης της εφαρμογής, δηλαδή της λίστας των video. Έχει αναφερθεί στην ενότητα της αρχικοποίησης ότι η λίστα χωράει 5 τίτλους video. Μετρίεται ο αριθμός των video, αν είναι μικρότερος του 5 δεν εμφανίζονται τα βελάκια up/down, αν είναι μεγαλύτερος τότε εμφανίζονται.

Κενό αρχείο videoList.xml



EIKONA 73



EIKONA 74

```
/*lista */
Display.setVideoList = function(nameList)
{
    var listHTML = "";

    var i=0;
    for (var name in nameList)
    {
        this.videoList[i] = document.getElementById("video"+i);
        listHTML = nameList[name] ;
        widgetAPI.putInnerHTML(this.videoList[i], listHTML);
        i++;
    }
    this.videoList[this.FIRSTIDX].style.backgroundImage= "url(Images/listBox/selector.png)";
    if(i>5)
    {
        document.getElementById("next").style.opacity = '1.0';
        document.getElementById("previous").style.opacity = '1.0';
    }
    listHTML = "1 / " + i;
    widgetAPI.putInnerHTML(document.getElementById("videoCount"), listHTML);
}
```

EIKONA 75

Ο χρήστης όπως έχει αναφερθεί έχει τη δυνατότητα μέσω του τηλεχειριστηρίου του να πλοηγείται στη λίστα των video και πρέπει να ξέρει σε ποιο Video είναι κάθε φορά. Για το λόγο αυτό στο επιλεγμένο video υπάρχει φόντο.



EIKONA 76

Για τη μετακίνηση στην λίστα υπάρχουν κάποιες περιπτώσεις που πρέπει να μελετηθούν:

- Η πρώτη και η πιο απλή, αν τα video είναι λιγότερα από 5 οπότε χωράνε όλα στην λίστα.
- Η δεύτερη είναι αν ο αριθμός των video είναι μεγαλύτερος από το 5, τότε έχουμε τις ακόλουθες υποπεριπτώσεις :
  1. Αν το επιλεγμένο video δεν είναι το τελευταίο και ο χρήστης δε θέλει να βρεθεί στο πρώτο video ή το επιλεγμένο video είναι το πρώτο και δεν θέλει ο χρήστης να πάει στο τελευταίο, δηλαδή να επιθυμεί να κινηθεί στα ενδιάμεσα.
  2. Το άλλο σενάριο είναι αν ο χρήστης βρίσκεται στο τελευταίο video και θέλει να πάει στο πρώτο.
  3. Το τελευταίο πιθανό σενάριο είναι ο χρήστης να βρίσκεται στο πρώτο video και να θέλει να πάει στο τελευταίο.

```

Display.setVideoListPosition = function(position, move)
{
    var listHTML = "";

    listHTML = (position + 1) + " / " + Data.getVideoCount();
    widgetAPI.putInnerHTML(document.getElementById("videoCount"), listHTML);

    if(Data.getVideoCount() < 5)
    {
        for (var i = 0; i < Data.getVideoCount(); i++)
        {
            if(i == position)
                this.videoList[i].style.backgroundImage= "url(Images/listBox/selector.png)";
            else
                this.videoList[i].style.backgroundImage= "url(none)";
        }
    }

    /* ploigisi sta endiamesa video ekto apo to teleutaio kai prwto gia perissotera apo 5 video */
    else if((this.currentWindow!=this.LASTIDX && move==Main.DOWN) || (this.currentWindow!=this.FIRSTIDX && move==Main.UP)
    {
        if(move == Main.DOWN)
            this.currentWindow ++;
        else
            this.currentWindow --;

        for (var i = 0; i <= this.LASTIDX; i++)
        {
            if(i == this.currentWindow)
                this.videoList[i].style.backgroundImage= "url(Images/listBox/selector.png)";
            else
                this.videoList[i].style.backgroundImage= "url(none)";
        }
    }

    /* gia na paei sto prwto tis listas pali*/
    else if(this.currentWindow == this.LASTIDX && move == Main.DOWN)
    {
        if(position == this.FIRSTIDX)
        {
            this.currentWindow = this.FIRSTIDX;

            for(i = 0; i <= this.LASTIDX; i++)
            {
                listHTML = Data.videoNames[i] ;
                widgetAPI.putInnerHTML(this.videoList[i], listHTML);

                if(i == this.currentWindow)
                    this.videoList[i].style.backgroundImage= "url(Images/listBox/selector.png)";
                else
                    this.videoList[i].style.backgroundImage= "url(none)";
            }
        }
        else
        {
            for(i = 0; i <= this.LASTIDX; i++)
            {
                listHTML = Data.videoNames[i + position - this.currentWindow] ;
                widgetAPI.putInnerHTML(this.videoList[i], listHTML);
            }
        }
    }

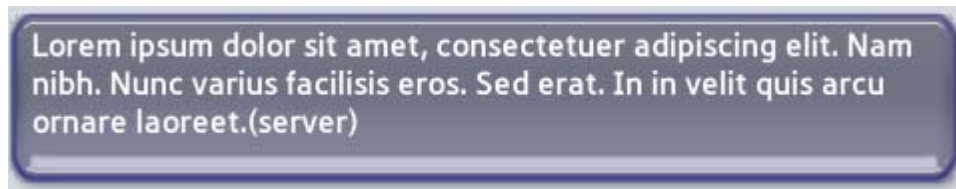
    /* gia na paei sto teleutaio tis listas pali */
    else if(this.currentWindow == this.FIRSTIDX && move == Main.UP)
    {
        if(position == Data.getVideoCount()-1)
        {
            this.currentWindow = this.LASTIDX;

            for(i = 0; i <= this.LASTIDX; i++)
            {
                listHTML = Data.videoNames[i + position - this.currentWindow] ;
                widgetAPI.putInnerHTML(this.videoList[i], listHTML);

                if(i == this.currentWindow)
                    this.videoList[i].style.backgroundImage= "url(Images/listBox/selector.png)";
                else
                    this.videoList[i].style.backgroundImage= "url(none)";
            }
        }
        else
        {
            for(i = 0; i <= this.LASTIDX; i++)
            {
                listHTML = Data.videoNames[i + position] ;
                widgetAPI.putInnerHTML(this.videoList[i], listHTML);
            }
        }
    }
}
}

```

Ολοκληρώνοντας γίνεται η εμφάνιση της σύντομης περιγραφής της παράστασης και η κατάσταση εμφάνισης ή απόκρυψης της εφαρμογής που καλείται στην main.



EIKONA 78

```
Display.setDescription = function(description)
{
    var descriptionElement = document.getElementById("description");

    widgetAPI.putInnerHTML(descriptionElement, description);
}

Display.hide = function()
{
    document.getElementById("main").style.display="none";
}

Display.show = function()
{
    document.getElementById("main").style.display="block";
}
```

EIKONA 79

## Common/API/ TVKeyValue.js

Το JavaScript αρχείο είναι βοηθητικό και προσδιορίζει τα πλήκτρα του τηλεχειριστηρίου που έχουν πατηθεί από το χρήστη.

```

*@file TVKeyValue.js
if (this.Common == null) {
    this.Common = new Object();
}
if (this.Common.API == null) {
    this.Common.API = new Object();
}

Common.API.TVKeyValue = function(){
    var $THIS$ = this;

    this.KEY_TOOLS = 75;
    this.KEY_MUTE = 27;
    this.KEY_RETURN = 88;
    this.KEY_UP = 29460;
    this.KEY_DOWN = 29461;
    this.KEY_LEFT = 4;
    this.KEY_RIGHT = 5;
    this.KEY_WHEELDOWN = 29469;
    this.KEY_WHEELUP = 29468;
    this.KEY_ENTER = 29443;
    this.KEY_INFO = 31;
    this.KEY_EXIT = 45;
    this.KEY_RED = 108;
    this.KEY_GREEN = 20;
    this.KEY_YELLOW = 21;
    this.KEY_BLUE = 22;
    this.KEY_INFOLINK = 147;
    this.KEY_RW = 69;
    this.KEY_PAUSE = 74;
    this.KEY_FF = 72;
    this.KEY_PLAY = 71;
    this.KEY_STOP = 70;
    this.KEY_1 = 101;
    this.KEY_2 = 98;
    this.KEY_3 = 6;
    this.KEY_4 = 8;
    this.KEY_5 = 9;
    this.KEY_6 = 10;
    this.KEY_7 = 12;
    this.KEY_8 = 13;
    this.KEY_9 = 14;
    this.KEY_0 = 17;
    this.KEY_EMPTY = 0;

    this.KEY_PRECH = 259;
    this.KEY_SOURCE = 222;
    this.KEY_CHLIST = 84;
    this.KEY_MENU = 262;
    this.KEY_ULINK = 115;
    this.KEY_CC = 118;
    this.KEY_CONTENT = 261;
    this.KEY_FAVCH = 256;
    this.KEY_REC = 192;
    this.KEY_EMODE = 148;
    this.KEY_DMA = 260;

    this.KEY_PANEL_CH_UP = 105;
    this.KEY_PANEL_CH_DOWN = 106;
    this.KEY_PANEL_VOL_UP = 203;
    this.KEY_PANEL_VOL_DOWN = 204;
    this.KEY_PANEL_ENTER = 309;
    this.KEY_PANEL_SOURCE = 612;
    this.KEY_PANEL_MENU = 613;
    this.KEY_PANEL_POWER = 614;

    // For all key regist

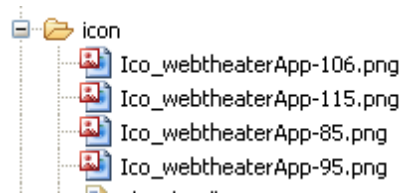
    this.KEY_POWER = 76;
    this.KEY_TV = 77;
    this.KEY_VOL_UP = 7;
    this.KEY_VOL_DOWN = 11;
    this.KEY_CH_UP = 68;
    this.KEY_CH_DOWN = 65;
    this.KEY_TTX_MIX = 650;
    this.KEY_GUIDE = 651;
    this.KEY_SUBTITLE = 652;
    this.KEY_ASPECT = 653;
    this.KEY_DOLBY_SFR = 654;
    this.KEY_MTS = 655;
    this.KEY_I2 = 1057;
    this.KEY_DISC_MENU = 1086;
    this.KEY_3D = 1219;
    this.KEY_PIP_CWOF = 1032;
    this.KEY_AD = 1039;
    this.KEY_PMODE = 1040;
    this.KEY_SMODE = 1043;
    this.KEY_PIP_CHUP = 1050;
    this.KEY_PIP_CHDOWN = 1051;
    this.KEY_FF_ = 1078;
    this.KEY_REWIND_ = 1080;
    this.KEY_SUB_TITLE = 1089;
    this.KEY_SLEEP = 1097;
    this.KEY_D_AUDIO = 1236;
    this.KEY_D_VIEW_MODE = 1249;
    this.KEY_STEP = 1023; // REC PAUSE(BD)
    this.KEY_CALLER_ID = 1128; // FULL SCREEN (BD)
    this.KEY_ZOOM1 = 1083;
}

```

## Φάκελοι

- **Ico**

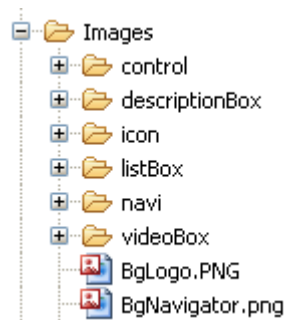
Εδώ βρίσκονται τα εικονίδια της εφαρμογής σε διάφορες διαστάσεις .



EIKONA 81

- **Images**

Εδώ βρίσκονται όλες οι εικόνες που χρησιμοποιεί η εφαρμογή .



EIKONA 82

- **Videos**

Ο φάκελος αυτός βρίσκεται στον server και έχει τα αρχεία video. Είναι ο φάκελος που διαβάζει το videoList.xml

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Από την έρευνα που πραγματοποιήσα διαπίστωσα τα εξής:

1. Τα τελευταία χρόνια οι υπηρεσίες των video on demand παρουσιάζουν μια ανοδική πορεία των υπηρεσιών τους και όλο και περισσότεροι χρήστες τις επιλέγουν ως τρόπο ψυχαγωγίας.
2. Το θεατρόφιλο κοινό είναι μεγάλο και το ενδιαφέρον του για παρακολούθηση θεατρικών παραστάσεων είναι αρκετά αυξημένο. Όμως αρκετές φορές η ζωντανή παρακολούθηση της παράστασης δεν είναι εφικτή λόγω διαφόρων εμποδίων που έχουν προαναφερθεί στην εισαγωγή .

Η εφαρμογή αυτή, δεν μπορεί βέβαια να υποκαταστήσει το συναίσθημα της «μέθεξης» της ζωντανής παράστασης, όμως αποτελεί μια υποσχόμενη μελλοντική υπηρεσία στο Home Entertainment.

Θα μπορούσε επιπλέον να συνδυαστεί με μια ιστοσελίδα δίνοντας τη δυνατότητα στους χρήστες-μέλη της να αποφασίζουν πού θέλουν να παρακολουθήσουν τη μαγνητοσκοπημένη παράσταση που τους ενδιαφέρει στον υπολογιστή ή στην τηλεόραση τους. Οι χρήστες θα επιλέγουν το χρόνο και τον τρόπο παρακολούθησης της μαγνητοσκοπημένης παράστασης και θα μπορούν να βιώνουν την χαρά και την ψυχαγωγία του θεάτρου και ας βρίσκονται στο προσωπικό τους χώρο.

## ΓΛΩΣΣΑΡΙ

- **SDK:** Software Development Kit
- **DOM :**Document Object Model
- **DHTML:** Dynamic HyperText Markup Language
- **AIR:** Adobe Integrated Runtime
- **Bpp:**bits per pixel
- **Common modules:** παρέχει πληροφορίες σχετικά με ορισμένα αντικείμενα γενικής χρήσης που μπορούν να χρησιμοποιηθούν από κάθε εφαρμογή.
- **Video playback :** χρησιμοποιείται για μια συσκευή που μπορεί να κρατήσει τα video στη μνήμη και στη συνέχεια να τα αναπαράγει.
- **Application manager** (Διαχειριστής Εφαρμογών): Παρέχει τα object modules που όλες οι εφαρμογές μπορούν να χρησιμοποιήσουν.
- **Common module:** επιτρέπουν στις εφαρμογές να πραγματοποιούν βασικές εργασίες για να τρέχουν κανονικά στην οθόνη.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- Television as an App: Taming the Device Fragmentation Problem  
<http://www.activevideo.com/file/1456/>
- HTML5 and Companion Web Technologies as a Universal Platform for Interactive Internet TV Advertising <http://www.chanthaburi.buu.ac.th/~worawit/seminar2556/53310276.pdf>
- Roles of Smart TV in IoT-environments: a Survey  
<http://www.fruct.org/publications/fruct13/files/Yus.pdf>
- Smart TV Hacking: Crash Testing Your Home Entertainment  
<http://www.codenomicon.com/resources/whitepapers/codenomicon-wp-smart-tv-fuzzing.pdf>
- Nielsen "Cross-Platform Report" September 2013  
<http://nielsen.com/content/dam/corporate/us/en/reports-downloads/2013%20Reports/Q2-2013-Cross-Platform-Report.pdf>
- Eurostat pocketbook on "Cultural Statistics" in April 2011,  
[http://epp.eurostat.ec.europa.eu/cache/ITY\\_OFFPUB/KS-32-10-374/EN/KS-32-10-374-EN.PDF](http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-32-10-374/EN/KS-32-10-374-EN.PDF)
- Αθηνόραμα 1/2 [http://www.athinorama.gr/MediaKit/users\\_a.aspx](http://www.athinorama.gr/MediaKit/users_a.aspx)
- Αθηνόραμα 2/2 [http://www.athinorama.gr/mediakit/statistics2013\\_a.aspx](http://www.athinorama.gr/mediakit/statistics2013_a.aspx)
- [http://en.wikipedia.org/wiki/Google\\_TV](http://en.wikipedia.org/wiki/Google_TV)
- <http://www.google.com/tv/apps.html>
- <http://www.chanthaburi.buu.ac.th/~worawit/seminar2556/53310276.pdf>
- <http://www.techopedia.com/>
- <http://en.wikipedia.org/>
- <http://searchtelecom.techtarget.com>
- <http://www.garymcgath.com>
- <http://windows.microsoft.com>
- <http://www.samsungdforum.com/MainGuide>

## ΠΑΡΑΡΤΗΜΑ -ΚΩΔΙΚΑΣ

### Config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.samsung.com/">
  <cpname itemType="string"/>
  <cplogo itemType="string"/>
  <cpauthjs itemType="string"/>
    <ThumbIcon itemType="string">icon/Ico_webtheaterApp-106.png</ThumbIcon>
    <BigThumbIcon itemType="string">icon/Ico_webtheaterApp-115.png</BigThumbIcon>
    <ListIcon itemType="string">icon/Ico_webtheaterApp-85.png</ListIcon>
    <BigListIcon itemType="string">icon/Ico_webtheaterApp-95.png</BigListIcon>
  <category itemType="string">Videos</category>
  <autoUpdate itemType="boolean">n</autoUpdate>
  <ver itemType="string">0.100</ver>
  <mgrver itemType="string"/>
    <fullwidget itemType="boolean">y</fullwidget>
  <movie itemType="boolean">y</movie>
  <type itemType="string">user</type>
  <srcctl itemType="boolean">y</srcctl>
  <ticker itemType="boolean">n</ticker>
  <childlock itemType="boolean">n</childlock>
    <videomute itemType="boolean">n</videomute>
    <audiomute itemType="boolean">n</audiomute>
  <dcont itemType="boolean">y</dcont>
  <widgetname itemType="string">WebTheaterPlayer</widgetname>
  <description itemType="string">Dimiourgia player gia to project webtheater.co sta plaisia tis diplwmatikis mou
  ergasias</description>
  <width itemType="string">960</width>
  <height itemType="string">540</height>
  <author itemType="group">
    <name itemType="string">Sofia Agraftoti</name>
    <email itemType="string">.</email>
    <link itemType="string">http://www.webtheater.co</link>
    <organization itemType="string">.</organization>
  </author>

</widget>
```

## Index.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/div/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>WebTheater player</title>

    <!-- Common widget API -->
    <script type='text/javascript' language='javascript' src='$MANAGER_WIDGET/Common/API/Widget.js'></script>
    <script type='text/javascript' language='javascript'
src='$MANAGER_WIDGET/Common/API/TVKeyValue.js'></script>
    <script type='text/javascript' language='javascript' src='$MANAGER_WIDGET/Common/API/Plugin.js'></script>

    <!-- Widget code -->
    <script language="javascript" type="text/javascript" src="Javascript/Main.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Server.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Data.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Player.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Display.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Audio.js"></script>

    <!-- Style sheets -->
    <link rel="stylesheet" href="CSS/Main.css" type="text/css">

    <!-- Plugins -->
    <object id="pluginPlayer" border=0 classid="clsid:SAMSUNG-INFOLINK-PLAYER" ></object>
    <object id="pluginTVMW" border=0 classid="clsid:SAMSUNG-INFOLINK-TVMW"></object>
    <object id="pluginAudio" border=0 classid="clsid:SAMSUNG-INFOLINK-AUDIO"></object>

  </head>

  <body onload="Main.onLoad();" onunload="Main.onUnload();">
    <!-- receive the key events -->
    <a href='javascript:void(0);' id='anchor' onkeydown='Main.keyDown();'></a>

    <!-- Layout -->
    <div id="main">
      <div id="logo"> <span id="text"> </img> </span> WEBTheater Player

      </div>

      <div id="leftHalf">
        <div id="videoList" class="style_videoList">
          <div id="video0"></div>
          <div id="video1"></div>
          <div id="video2"></div>
          <div id="video3"></div>
          <div id="video4"></div>

          <div id="videoCount"></div>
        </div>
        <div id="previous"></div>
        <div id="next"></div>
      </div>
    </body>
  </html>
```



## videoList.xml

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <item>
      <title>Video1</title>
      <link>http://galaxy.hua.gr/~itp12201/videos/movie.mp4</link>
      <description>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Nam nibh. Nunc varius facilisis eros.
        Sed erat. In in velit quis arcu ornare laoreet.(server)
      </description>
    </item>

    <item>
      <title>Video2</title>
      <link>http://galaxy.hua.gr/~itp12201/videos/sirano_pallas.mp4</link>
      <description>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Nam nibh. Nunc varius facilisis eros.
        Sed erat. In in velit quis arcu ornare laoreet.(server)
      </description>
    </item>

  </channel>

</rss>
```

## Main.css

```
/* Defaults */

*
{
  padding: 0;
  margin: 0;
  border: 0;
  position: relative;
  color: #fff;
  background-color: transparent;
  font-family: Verdana;
}

/* Layout */
body
{
  width: 960px;
  height: 540px;
```

```

}

#main
{
    position: absolute;
    left: 0px; top: 0px;
    width: 960px; height: 540px;

}

#logo
{
    position: absolute;
    top: 10px;
    height: 45px;
    width: 100%;
    background-image: url("../Images/BgLogo.png");
    background-repeat: no-repeat;
    font-size: 25px;
    padding-top: 4px;
    padding-left: 20px;
    margin-left: 10px;
    margin-right: 5px;
    z-index: 10;

}

#text {
    font-size: 15px;

}

/* Right Half */
#rightHalf
{
    position: absolute;
    left: 420px; top: -5px;
    width: 540px; height: 540px;
    background-image: url("../Images/videoBox/rightHalf.png");
}

#videoBox_top
{
    position: absolute;
    left: 30px; top: 55px;
    width: 488px; height: 8px;
    background-image: url("../Images/videoBox/top.png");
}

#videoBox_left
{
    position: absolute;
    left: 30px; top: 55px;
    width: 8px; height: 278px;
    background-image: url("../Images/videoBox/left.png");
}

```

```

}

#videoBox_right
{
    position: absolute;
    left: 520px; top: 55px;
    width: 8px; height: 278px;
    background-image: url("../Images/videoBox/right.png");
}

#videoBox_bottom
{
    position: absolute;
    left: 30px; top: 333px;
    width: 498px; height: 56px;
    background-image: url("../Images/videoBox/bottom.png");
    -webkit-box-shadow: 2px 10px 8px -8px rgba(0,0,0,0.58);
    -moz-box-shadow: 2px 10px 8px -8px rgba(0,0,0,0.58);
    box-shadow: 2px 10px 8px -8px rgba(0,0,0,0.58);
}

#time
{
    position: absolute;
    left: 8px; top: 0px;
    width: 472px; height: 24px;
}

#progressBarBG
{
    position: absolute;
    left: 4px; top: 8px;
    width: 400px; height: 8px;
    background-image: url("../Images/videoBox/progressBarBG.png");
}

#progressBar
{
    position: absolute;
    left: 0px; top: 0px;
    width: 0%; height: 8px;
    background-image: url("../Images/videoBox/progressBar.png");
}

#timeInfo
{
    position: absolute;
    left: 408px; top: 4px;
    width: 60px; height: 16px;

    text-align: center;
    font-size: 12px;
}

#volume
{
    position: absolute;
    left: 8px; top: 24px;

```

```

    width:472px; height:24px;
}

#volumeIcon
{
    position:absolute;
    left:4px; top:4px;
    width:17px; height:16px;
    background-image:url("../Images/videoBox/volume.png");
}

#volumeBarBG
{
    position:absolute;
    left:22px; top:8px;
    width:144px; height:8px;
    background-image:url("../Images/videoBox/volumeBarBG.png");
}

#volumeBar
{
    position:absolute;
    left:0px; top:0px;
    width:0%; height:8px;
    background-image:url("../Images/videoBox/volumeBar.png");
}

#volumeInfo
{
    position:absolute;
    left:168px; top:4px;
    width:32px; height:16px;

    text-align:left;
    font-size:12px;
}

#videoControl
{
    position:absolute;
    left:250px; top:25px;
    width:100px; height:20px;
}

#rewind
{
    position:absolute;
    left:0px; top:0px;
    width:20px; height:20px;
    background-image:url("../Images/control/krew.png");
    opacity:0.2;
}

#play
{
    position:absolute;
    left:20px; top:0px;
    width:20px; height:20px;
}

```

```

background-image:url("../Images/control/kplay.png");
opacity:1.0;
}

#stop
{
    position:absolute;
    left:40px; top:0px;
    width:20px; height:20px;
    background-image:url("../Images/control/kstop.png");
    opacity:0.2;
}

#pause
{
    position:absolute;
    left:60px; top:0px;
    width:20px; height:20px;
    background-image:url("../Images/control/kpause.png");
    opacity:0.2;
}

#forward
{
    position:absolute;
    left:80px; top:0px;
    width:20px; height:20px;
    background-image:url("../Images/control/kfwd.png");
    opacity:0.2;
}

#status
{
    position:absolute;
    left:390px; top:28px;
    width: 98px; height:16px;

    text-align:left;
    font-size:12px;
}

#images
{
    width  : 120px;
    height : 25px;
    border  : 1px solid #FFC71E;
    float  : left;
}

#description_top
{
    position:absolute;
    left:30px; top:400px;
    width:488px; height:90px;
    background-image:url("../Images/descriptionBox/description_top.png");
}

#description_bottom
{
    position:absolute;

```

```

    left:30px; top:440px;
    width:488px;
}

#description
{
    position:absolute;
    left:45px; top:410px;
    width:458px; height:80px;

    text-align:left;
    font-size:16px;
    overflow:hidden;
    text-overflow : ellipsis;
}

/* Left Half */
#leftHalf
{
    position:absolute;
    left: 0px; top:0px;
    width:420px; height:540px;
    background-color: #d4dadf;
}

#videoList
{
    position:absolute;
    left:20px; top:65px;
    width: 390px; height:440px;
    background-image:url("../Images/listBox/listBox.png");
    background-repeat:no-repeat;
}

#video0
{
    position:absolute;
    left:30px; top:70px;
    width:330px; height:47px;

    background-repeat:no-repeat;
    padding-left:20px;
    padding-top:10px;
}

#video1
{
    position:absolute;
    left:30px; top:130px;
    width:330px; height:47px;

    background-repeat:no-repeat;
    padding-left:20px;
    padding-top:10px;
}

#video2

```

```

{
    position: absolute;
    left: 30px; top: 190px;
    width: 330px; height: 47px;

    background-repeat: no-repeat;
    padding-left: 20px;
    padding-top: 10px;
}

#video3
{
    position: absolute;
    left: 30px; top: 250px;
    width: 330px; height: 47px;

    background-repeat: no-repeat;
    padding-left: 20px;
    padding-top: 10px;
}

#video4
{
    position: absolute;
    left: 30px; top: 310px;
    width: 330px; height: 47px;

    background-repeat: no-repeat;
    padding-left: 20px;
    padding-top: 10px;
}

#videoCount
{
    position: absolute;
    left: 330px; top: 30px;
    width: 30px; height: 20px;

    text-align: left;
    font-size: 12px;
}

#previous
{
    position: absolute;
    left: 210px; top: 80px;
    width: 30px; height: 30px;
    background-image: url("../Images/listBox/previous.png");
    opacity: 0.2;
}

#next
{
    position: absolute;
    left: 210px; top: 430px;
    width: 30px; height: 30px;
    background-image: url("../Images/listBox/next.png");
    opacity: 0.2;
}

```

```

}

.style_videoList
{
    color:#FFFFFF;
    text-align:left;
    font-size:20px;
}

#navi
{
    position: absolute;
    top: 495px;
    height: 50px;
    width: 100%;
    background-image: url("../Images/BgNavigator.png");
    background-repeat:no-repeat;
    padding-left:20px;
    margin-left:10px;
    margin-right:5px;
}

#help_navi
{
    position:absolute;
    left:25px; top:5px;
    width:885px; height:28px;
}
#help_navi
{
    font-size:14px;
}
.style_navi {
    bottom:4px;
    height:20px;
    font-size:20px;

}

```

## **TVKeyValue.js**

```

/**
 *@file    TVKeyValue.js
 *@brief   TV key
 *@author   samsung

 */
if (this.Common == null) {
    this.Common = new Object();
}
if (this.Common.API == null) {
    this.Common.API = new Object();
}

```

}

```
Common.API.TVKeyValue = function(){  
    var $THIS$ = this;
```

```
  
    this.KEY_TOOLS = 75;  
    this.KEY_MUTE = 27;  
    this.KEY_RETURN = 88;  
    this.KEY_UP = 29460;  
    this.KEY_DOWN = 29461;  
    this.KEY_LEFT = 4;  
    this.KEY_RIGHT = 5;  
    this.KEY_WHEELDOWN = 29469;  
    this.KEY_WHEELUP = 29468;  
    this.KEY_ENTER = 29443;  
    this.KEY_INFO = 31;  
    this.KEY_EXIT = 45;  
    this.KEY_RED = 108;  
    this.KEY_GREEN = 20;  
    this.KEY_YELLOW = 21;  
    this.KEY_BLUE = 22;  
    this.KEY_INFOLINK = 147;  
    this.KEY_RW = 69;  
    this.KEY_PAUSE = 74;  
    this.KEY_FF = 72;  
    this.KEY_PLAY = 71;  
    this.KEY_STOP = 70;  
    this.KEY_1 = 101;  
    this.KEY_2 = 98;  
    this.KEY_3 = 6;  
    this.KEY_4 = 8;  
    this.KEY_5 = 9;  
    this.KEY_6 = 10;  
    this.KEY_7 = 12;  
    this.KEY_8 = 13;  
    this.KEY_9 = 14;  
    this.KEY_0 = 17;  
    this.KEY_EMPTY = 0;  
  
    this.KEY_PRECH = 259;  
    this.KEY_SOURCE = 222;  
    this.KEY_CHLIST = 84;  
    this.KEY_MENU = 262;  
    this.KEY_WLINK = 115;  
    this.KEY_CC = 118;  
    this.KEY_CONTENT = 261;  
    this.KEY_FAVCH = 256;  
    this.KEY_REC = 192;  
    this.KEY_EMODE = 148;  
    this.KEY_DMA = 260;  
  
    this.KEY_PANEL_CH_UP = 105;  
    this.KEY_PANEL_CH_DOWN = 106;  
    this.KEY_PANEL_VOL_UP = 203;  
    this.KEY_PANEL_VOL_DOWN = 204;  
    this.KEY_PANEL_ENTER = 309;  
    this.KEY_PANEL_SOURCE = 612;
```

```

this.KEY_PANEL_MENU = 613;
this.KEY_PANEL_POWER = 614;

```

```

// For all key regist

```

```

this.KEY_POWER = 76;
this.KEY_TV = 77;
this.KEY_VOL_UP = 7;
this.KEY_VOL_DOWN = 11;
this.KEY_CH_UP = 68;
this.KEY_CH_DOWN = 65;
this.KEY_TTX_MIX = 650;
this.KEY_GUIDE = 651;
this.KEY_SUBTITLE = 652;
this.KEY_ASPECT = 653;
this.KEY_DOLBY_SRR = 654;
this.KEY_MTS = 655;
this.KEY_12 = 1057;
this.KEY_DISC_MENU = 1086;
this.KEY_3D = 1219;
this.KEY_PIP_ONOFF = 1032;
this.KEY_AD = 1039;
this.KEY_PMODE = 1040;
this.KEY_SMODE = 1043;
this.KEY_PIP_CHUP = 1050;
this.KEY_PIP_CHDOWN = 1051;
this.KEY_FF = 1078;
this.KEY_REWIND = 1080;
this.KEY_SUB_TITLE = 1089;
this.KEY_SLEEP = 1097;
this.KEY_D_AUDIO = 1236;
this.KEY_D_VIEW_MODE = 1249;
this.KEY_STEP = 1023;
this.KEY_CALLER_ID = 1128;
this.KEY_ZOOM1 = 1083;

```

```

// REC PAUSE(BD)
// FULL SCREEN (BD)

```

```

}

```

## Audio.js

```

var Audio =
{
  plugin : null
}

```

```

Audio.init = function()
{
  var success = true;

  this.plugin = document.getElementById("pluginAudio");

  if (!this.plugin)
  {
    success = false;
  }

  return success;
}

```

```

}

Audio.setRelativeVolume = function(vol)
{
    this.plugin.SetVolumeWithKey(vol);
    Display.setVolume( this.getVolume() );
}

Audio.getVolume = function()
{
    alert("Volume : " + this.plugin.GetVolume());
    return this.plugin.GetVolume();
}

```

## **Data.js**

```

var Data =
{
    videoNames : [ ],
    videoURLs : [ ],
    videoDescriptions : [ ],
}

Data.setVideoNames = function(list)
{
    this.videoNames = list;
}

Data.setVideoURLs = function(list)
{
    this.videoURLs = list;
    alert("video urls:" +this.videoURLs +" ");
}

Data.setVideoDescriptions = function(list)
{
    this.videoDescriptions = list;
    alert("Description:" +this.videoDescriptions);
}

Data.getVideoURL = function(index)
{
    var url = this.videoURLs[index];

    if (url)  // Check for undefined entry
    {
        alert("URL is: "+url)
        return url;
    }
    else
    {
        return null;
    }
}

```

```

Data.getVideoCount = function()
{
    return this.videoURLs.length;
}

Data.getVideoNames = function()
{
    return this.videoNames;
}

Data.getVideoDescription = function(index)
{
    var description = this.videoDescriptions[index];

    if (description) // Check for undefined entry
    {
        return description;
    }
    else
    {
        return "No description";
    }
}

```

### Display.js

```

var Display =
{
    statusDiv : null,      /* css class status */
    FIRSTIDX : 0,          /* first video */
    LASTIDX : 4,           /* last video */
    currentWindow : 0,

    SELECTOR : 0, /* koumpia up /down */
    LIST : 1,     /* arithmos 1ou video */

    videoList : new Array()
}

Display.init = function()
{
    var success = true;

    this.statusDiv = document.getElementById("status");

    if (!this.statusDiv)
    {
        success = false;
    }

    return success;
}

Display.status = function(status)
{

```

```

        alert(status);
        widgetAPI.putInnerHTML(this.statusDiv, status);
    }

    /* ypologismos diarkeias */
    Display.setTotalTime = function(total)
    {
        this.totalTime = total;
    }
    Display.setTime = function(time)
    {
        var timePercent = (100 * time) / this.totalTime;
        var timeElement = document.getElementById("timeInfo");
        var timeHTML = "";
        var timeHour = 0; var timeMinute = 0; var timeSecond = 0;
        var totalTimeHour = 0; var totalTimeMinute = 0; var totalTimeSecond = 0;

        document.getElementById("progressBar").style.width = timePercent + "%";

        if(Player.state == Player.PLAYING)
        {
            totalTimeHour = Math.floor(this.totalTime/3600000);
            timeHour = Math.floor(time/3600000);

            totalTimeMinute = Math.floor((this.totalTime%3600000)/60000);
            timeMinute = Math.floor((time%3600000)/60000);

            totalTimeSecond = Math.floor((this.totalTime%60000)/1000);
            timeSecond = Math.floor((time%60000)/1000);

            timeHTML = timeHour + ":";

            if(timeMinute == 0)
                timeHTML += "00:";
            else if(timeMinute <10)
                timeHTML += "0" + timeMinute + ":";
            else
                timeHTML += timeMinute + ":";

            if(timeSecond == 0)
                timeHTML += "00/";
            else if(timeSecond <10)
                timeHTML += "0" + timeSecond + "/";
            else
                timeHTML += timeSecond + "/";

            timeHTML += totalTimeHour + ":";

            if(totalTimeMinute == 0)
                timeHTML += "00:";
            else if(totalTimeMinute <10)
                timeHTML += "0" + totalTimeMinute;
            else
                timeHTML += totalTimeMinute;

            if(totalTimeSecond == 0)
                timeHTML += "00";
            else if(totalTimeSecond <10)

```



```

        {
            if(i == position)
                this.videoList[i].style.backgroundImage=
"url(Images/listBox/selector.png)";
            else
                this.videoList[i].style.backgroundImage= "url(none)";
        }
    }

    /* ploigisi sta endiamesa video ektos apo to teleutaio kai prwto gia
perissotera apo 5 video */
    else if((this.currentWindow!=this.LASTIDX && move==Main.DOWN) ||
(this.currentWindow!=this.FIRSTIDX && move==Main.UP))
    {
        if(move == Main.DOWN)
            this.currentWindow ++;
        else
            this.currentWindow --;

        for (var i = 0; i <= this.LASTIDX; i++)
        {
            if(i == this.currentWindow)
                this.videoList[i].style.backgroundImage=
"url(Images/listBox/selector.png)";
            else
                this.videoList[i].style.backgroundImage= "url(none)";
        }
    }
    /* gia na paei sto prwto tis listas pali*/
    else if(this.currentWindow == this.LASTIDX && move == Main.DOWN)
    {
        if(position == this.FIRSTIDX)
        {
            this.currentWindow = this.FIRSTIDX;

            for(i = 0; i <= this.LASTIDX; i++)
            {
                listHTML = Data.videoNames[i] ;
                widgetAPI.putInnerHTML(this.videoList[i], listHTML);

                if(i == this.currentWindow)
                    this.videoList[i].style.backgroundImage=
"url(Images/listBox/selector.png)";
                else
                    this.videoList[i].style.backgroundImage= "url(none)";
            }
        }
        else
        {
            for(i = 0; i <= this.LASTIDX; i++)
            {
                listHTML = Data.videoNames[i + position - this.currentWindow] ;
                widgetAPI.putInnerHTML(this.videoList[i], listHTML);
            }
        }
    }
    /* gia na paei sto teleutaio tis listas pali */
    else if(this.currentWindow == this.FIRSTIDX && move == Main.UP)
    {

```

```

    if(position == Data.getVideoCount()-1)
    {
        this.currentWindow = this.LASTIDX;

        for(i = 0; i <= this.LASTIDX; i++)
        {
            listHTML = Data.videoNames[i + position - this.currentWindow] ;
            widgetAPI.putInnerHTML(this.videoList[i], listHTML);

            if(i == this.currentWindow)
                this.videoList[i].style.backgroundImage=
"url(Images/listBox/selector.png)";
            else
                this.videoList[i].style.backgroundImage= "url(none)";
        }
    }
    else
    {
        for(i = 0; i <= this.LASTIDX; i++)
        {
            listHTML = Data.videoNames[i + position] ;
            widgetAPI.putInnerHTML(this.videoList[i], listHTML);
        }
    }
}

/* emfanisis perigrafis kai perivalontos */
Display.setDescription = function(description)
{
    var descriptionElement = document.getElementById("description");

    widgetAPI.putInnerHTML(descriptionElement, description);
}

Display.hide = function()
{
    document.getElementById("main").style.display="none";
}

Display.show = function()
{
    document.getElementById("main").style.display="block";
}

```

### Main.js

```

var widgetAPI = new Common.API.Widget();
var tvKey = new Common.API.TVKeyValue();

var Main =
{
    selectedVideo : 0,
    mode : 0,
    mute : 0,

    UP : 0,

```

```

    DOWN : 1,

    WINDOW : 0,
    FULLSCREEN : 1,

    NMUTE : 0,
    YMUTE : 1
}

Main.onLoad = function()
{
    if ( Player.init() && Audio.init() && Display.init() && Server.init() )
    {
        Display.setVolume( Audio.getVolume() );
        Display.setTime(0);

        Player.stopCallback = function()
        {
            /* Return to windowed mode when video is stopped
               (by choice or when it reaches the end) */
            Main.setWindowMode();
        }

        // Start retrieving data from server
        Server.dataReceivedCallback = function()
        {
            /* Use video information when it has arrived */
            Display.setVideoList( Data.getVideoNames() );
            Main.updateCurrentVideo();
        }

        Server.fetchVideoList(); /* Request video information from server */

        // Enable key event processing
        this.enableKeys();

        widgetAPI.sendReadyEvent();
    }
    else
    {
        alert("Failed to initialise");
    }
    var WIDGET_ID = curWidget.id;

    alert("#WIDGET_ID# =" + WIDGET_ID);
}

showHandler = function()
{
    alert("Show handler");
}
Main.onUnload = function()
{
    Player.deinit();
}

Main.updateCurrentVideo = function(move)
{
    Player.setVideoURL( Data.getVideoURL(this.selectedVideo) );
}

```

```

Display.setVideoListPosition(this.selectedVideo);

Display.setVideoListPosition(this.selectedVideo, move);

Display.setDescription( Data.getVideoDescription(this.selectedVideo));
}

Main.enableKeys = function()
{
    document.getElementById("anchor").focus();
}

Main.keyDown = function()
{
    var keyCode = event.keyCode;
    alert("Key pressed: " + keyCode);

    switch(keyCode)
    {
        case tvKey.KEY_RETURN:
        case tvKey.KEY_PANEL_RETURN:
            alert("RETURN");
            Player.stopVideo();
            widgetAPI.sendReturnEvent();
            break;

        case tvKey.KEY_PLAY:
            alert("PLAY");

            this.handlePlayKey();
            break;

        case tvKey.KEY_STOP:
            alert("STOP");
            Player.stopVideo();
            break;

        case tvKey.KEY_PAUSE:
            alert("PAUSE");
            this.handlePauseKey();
            break;

        case tvKey.KEY_FF:
            alert("FF");
            if(Player.getState() != Player.PAUSED)
                Player.skipForwardVideo();
            break;

        case tvKey.KEY_RW:
            alert("RW");
            if(Player.getState() != Player.PAUSED)
                Player.skipBackwardVideo();
            break;

        case tvKey.KEY_VOL_UP:
        case tvKey.KEY_PANEL_VOL_UP:
            alert("VOL_UP");

```

```

        if(this.mute == 0)
            Audio.setRelativeVolume(0);
        break;

    case tvKey.KEY_VOL_DOWN:
    case tvKey.KEY_PANEL_VOL_DOWN:
        alert("VOL_DOWN");
        if(this.mute == 0)
            Audio.setRelativeVolume(1);
        break;

    case tvKey.KEY_DOWN:
        alert("DOWN");
        this.selectNextVideo(this.DOWN);
        break;

    case tvKey.KEY_UP:
        alert("UP");
        this.selectPreviousVideo(this.UP);
        break;

    case tvKey.KEY_ENTER:
    case tvKey.KEY_PANEL_ENTER:
        alert("ENTER");
        this.toggleMode();
        break;

    case tvKey.KEY_MUTE:
        alert("MUTE");
        this.muteMode();
        break;

    default:
        alert("Unhandled key");
        break;
    }
}

Main.handlePlayKey = function()
{
    switch ( Player.getState() )
    {
        case Player.STOPPED:
            Player.playVideo();
            break;

        case Player.PAUSED:
            Player.resumeVideo();
            break;

        default:
            alert("Ignoring play key, not in correct state");
            break;
    }
}

Main.handlePauseKey = function()
{
    switch ( Player.getState() )

```

```

    {
        case Player.PLAYING:
            Player.pauseVideo();
            break;

        default:
            alert("Ignoring pause key, not in correct state");
            break;
    }
}

Main.selectNextVideo = function(down)
{
    Player.stopVideo();

    this.selectedVideo = (this.selectedVideo + 1) % Data.getVideoCount();

    this.updateCurrentVideo(down);
}

Main.selectPreviousVideo = function(up)
{
    Player.stopVideo();

    if (--this.selectedVideo < 0)
    {
        this.selectedVideo += Data.getVideoCount();
    }

    this.updateCurrentVideo(up);
}

Main.setFullScreenMode = function()
{
    if (this.mode != this.FULLSCREEN)
    {
        Display.hide();

        Player.setFullscreen();

        this.mode = this.FULLSCREEN;
    }
}

Main.setWindowMode = function()
{
    if (this.mode != this.WINDOW)
    {
        Display.show();

        Player.setWindow();

        this.mode = this.WINDOW;
    }
}

Main.toggleMode = function()
{
    if(Player.getState() == Player.PAUSED)

```

```

    {
        Player.resumeVideo();
    }
    switch (this.mode)
    {
        case this.WINDOW:
            this.setFullScreenMode();
            break;

        case this.FULLSCREEN:
            this.setWindowMode();
            break;

        default:
            alert("ERROR: unexpected mode in toggleMode");
            break;
    }
}

Main.setMuteMode = function()
{
    if (this.mute != this.YMUTE)
    {
        var volumeElement = document.getElementById("volumeInfo");
        //Audio.plugin.SetSystemMute(true);
        Audio.plugin.SetUserMute(true);
        document.getElementById("volumeBar").style.backgroundImage =
"url(Images/videoBox/muteBar.png)";
        document.getElementById("volumeIcon").style.backgroundImage =
"url(Images/videoBox/mute.png)";
        widgetAPI.putInnerHTML(volumeElement, "MUTE");
        this.mute = this.YMUTE;
    }
}

Main.noMuteMode = function()
{
    if (this.mute != this.NMUTE)
    {
        Audio.plugin.SetUserMute(false);
        document.getElementById("volumeBar").style.backgroundImage =
"url(Images/videoBox/volumeBar.png)";
        document.getElementById("volumeIcon").style.backgroundImage =
"url(Images/videoBox/volume.png)";
        Display.setVolume( Audio.getVolume() );
        this.mute = this.NMUTE;
    }
}

Main.muteMode = function()
{
    switch (this.mute)
    {
        case this.NMUTE:
            this.setMuteMode();
            break;

        case this.YMUTE:

```

```

        this.noMuteMode();
        break;

    default:
        alert("ERROR: unexpected mode in muteMode");
        break;
    }
}

```

### Player.js

```

var Player =
{
    plugin : null,
    state : -1,          /* arxiki timi */
    skipState : -1,
    stopCallback : null, /* Callback function */
    originalSource : null,

    /* koumplia player */
    STOPPED : 0,
    PLAYING : 1,
    PAUSED : 2,
    FORWARD : 3,
    REWIND : 4
}

Player.init = function()
{
    var success = true;
    alert("player success : " + success);
    this.state = this.STOPPED;

    this.plugin = document.getElementById("pluginPlayer");
    alert("plugin"+this.plugin);
    if (!this.plugin)
    {
        alert("success this.plugin : " + success);
        success = false;
    }

    alert("success : " + success);

    this.setWindow();

    alert("success : " + success);

    this.plugin.OnCurrentPlayTime = 'Player.setCurTime';
    this.plugin.OnStreamInfoReady = 'Player.setTotalTime';
    this.plugin.OnBufferingStart = 'Player.onBufferingStart';
    this.plugin.OnBufferingProgress = 'Player.onBufferingProgress';
    this.plugin.OnBufferingComplete = 'Player.onBufferingComplete';

    alert("success : " + success);
    return success;
}

```

```

Player.deinit = function()
{
    alert("Player deinit !!! " );

    if (this.plugin)
    {
        this.plugin.Stop();
    }
}

Player.setWindow = function()
{
    alert("setting window");
    this.plugin.SetDisplayArea(458, 58, 472, 270);
}

Player.setFullscreen = function()
{
    this.plugin.SetDisplayArea(0, 0, 960, 540);
}

Player.setVideoURL = function(url)
{
    this.url = url;
    alert("URL = " + this.url);
}

Player.playVideo = function()
{
    if (this.url == null)
    {
        alert("No videos to play");
    }
    else
    {

        this.state = this.PLAYING;

        /* css value 0.2 for not press, 1.0 for press*/

        document.getElementById("play").style.opacity = '0.2';
        document.getElementById("stop").style.opacity = '1.0';
        document.getElementById("pause").style.opacity = '1.0';
        document.getElementById("forward").style.opacity = '1.0';
        document.getElementById("rewind").style.opacity = '1.0';
        Display.status("---Play---");
        this.plugin.Play(this.url);

    }
}

Player.pauseVideo = function()
{
    this.state = this.PAUSED;
    document.getElementById("play").style.opacity = '1.0';
    document.getElementById("stop").style.opacity = '1.0';
    document.getElementById("pause").style.opacity = '0.2';
    document.getElementById("forward").style.opacity = '0.2';
    document.getElementById("rewind").style.opacity = '0.2';
}

```

```

        Display.status("---Pause---");
        this.plugin.Pause();
    }

    Player.stopVideo = function()
    {
        if (this.state != this.STOPPED)
        {
            this.state = this.STOPPED;
            document.getElementById("play").style.opacity = '1.0';
            document.getElementById("stop").style.opacity = '0.2';
            document.getElementById("pause").style.opacity = '0.2';
            document.getElementById("forward").style.opacity = '0.2';
            document.getElementById("rewind").style.opacity = '0.2';
            Display.status("---Stop---");
            this.plugin.Stop();
            Display.setTime(0);

            if (this.stopCallback)
            {
                this.stopCallback();
            }
        }
        else
        {
            alert("Ignoring stop request, not in correct state");
        }
    }

    Player.resumeVideo = function()
    {
        this.state = this.PLAYING;
        document.getElementById("play").style.opacity = '0.2';
        document.getElementById("stop").style.opacity = '1.0';
        document.getElementById("pause").style.opacity = '1.0';
        document.getElementById("forward").style.opacity = '1.0';
        document.getElementById("rewind").style.opacity = '1.0';
        Display.status("---Resume---");
        this.plugin.Resume();
    }

    Player.skipForwardVideo = function()
    {
        this.skipState = this.FORWARD;
        this.plugin.JumpForward(3);
    }

    Player.skipBackwardVideo = function()
    {
        this.skipState = this.REWIND;
        this.plugin.JumpBackward(3);
    }

    Player.getState = function()
    {
        return this.state;
    }

```

// Oi leitourgies forward kai backforward einai energeis mono otan exei ksekinisei to buffering

```
Player.onBufferingStart = function()
{
    Display.status("Buffering...");
    switch(this.skipState)
    {
        case this.FORWARD:
            document.getElementById("forward").style.opacity = '0.2';
            break;

        case this.REWIND:
            document.getElementById("rewind").style.opacity = '0.2';
            break;
    }
}
```

```
Player.onBufferingProgress = function(percent)
{
    var fileUrl =this.url;

    alert(fileUrl);
    Display.status("Buffering:" + percent + "%");
}
```

```
Player.onBufferingComplete = function()
{
    Display.status("Play");
    switch(this.skipState)
    {
        case this.FORWARD:
            document.getElementById("forward").style.opacity = '1.0';
            break;

        case this.REWIND:
            document.getElementById("rewind").style.opacity = '1.0';
            break;
    }
}
```

```
stopPlayer = function()
{
    Player.stopVideo();
}
```

```
Player.setCurTime = function(time)
{
    Display.setTime(time);
}
```

```
Player.setTotalTime = function()
{
    Display.setTotalTime(Player.plugin.GetDuration());
}
```

```

/* Diaforoi elexnoi gia ton server to diktio tou bandwidth */
onServerError = function()
{
    Display.status("Server Error!");
}

OnNetworkDisconnected = function()
{
    Display.status("Network Error!");
}

getBandwidth = function(bandwidth) { alert("getBandwidth " + bandwidth); }

onDecoderReady = function() { alert("onDecoderReady"); }

/* elexnos apodosois
    Unsupported container
    Unsupported video codec
    Unsupported audio codec
    Unsupported video resolution
*/
onRenderError = function() { alert("onRenderError"); }

/* Buffering*/
setTottalBuffer = function(buffer) { alert("setTottalBuffer " + buffer); }

setCurBuffer = function(buffer) { alert("setCurBuffer " + buffer); }

```

### Server.js

```

var Server =
{
    /* Callback function to be set by client */
    dataReceivedCallback : null,

    XHRObj : null,
    url : "http://galaxy.hua.gr/~itp12201/XML/videoList.xml"
}

Server.init = function()
{
    var success = true;

    if (this.XHRObj)
    {
        this.XHRObj.destroy(); // epituximeni epikoinonia apodesmeusi tou
        antikeimenoy
        this.XHRObj = null;
    }

    return success;
}

```

```

/* function pou fernei tin lista */
Server.fetchVideoList = function()
{
    if (this.XHRobj == null)
    {
        this.XHRobj = new XMLHttpRequest();
    }

    if (this.XHRobj)
    {
        this.XHRobj.onreadystatechange = function()
        {
            if (Server.XHRobj.readyState == 4 && Server.XHRobj.status == 200
)
                {
                    Server.createVideoList();
                }

            this.XHRobj.open("GET", this.url, true);
            this.XHRobj.send(null);
        }
    }
    else
    {
        alert("Failed to create XHR");
    }
}

Server.createVideoList = function()
{
    if (this.XHRobj.status != 200)
    {
        Display.status("XML Server Error " + this.XHRobj.status);
    }
    else
    {
        var xmlElement = this.XHRobj.responseXML.documentElement;

        if (!xmlElement)
        {
            alert("Failed to get valid XML");
        }
        else
        {
            // Get all "item" elements
            var items = xmlElement.getElementsByTagName("item");

            var videoNames = [ ];
            var videoURLs = [ ];
            var videoDescriptions = [ ];

            for (var index = 0; index < items.length; index++)
            {
                var titleElement =
items[index].getElementsByTagName("title")[0];
                var descriptionElement =
items[index].getElementsByTagName("description")[0];

```

```

        var linkElement = items[index].getElementsByTagName("link")[0];

        if (titleElement && descriptionElement && linkElement)
        {
            videoNames[index] = titleElement.firstChild.data;
            videoURLs[index] = linkElement.firstChild.data;
            videoDescriptions[index] =
descriptionElement.firstChild.data;
        }

        Data.setVideoNames(videoNames);
        Data.setVideoURLs(videoURLs);
        Data.setVideoDescriptions(videoDescriptions);

        if (this.dataReceivedCallback)
        {
            this.dataReceivedCallback();    /* enimerosi oti ola ta data
            */
        }
    }
}

```