

Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Graduat en Mitjans Audiovisuals

PORTAFOLI DIGITAL MULTIMÈDIA

Memòria

JOSEP BORRÀS YÁÑEZ
PONENT: LINA JUAN

PRIMAVERA 2011



TecnoCampus
Mataró-Maresme

Dedicatòria

Per a Ximena i Salvador i tots aquests anys de suport.

Agraïments

Especial agraïment als professorsina que la dedicatòria.

Resum

Aquest projecte té com a objectiu la producció d'una aplicació informàtica amb contingut audiovisual, que permeti comunicar la informació desitjada mitjançant recursos interactius entre programa i usuari. Aquest producte digital s'anomena Portafoli Digital Multimèdia. Una aplicació on plasmar el treball propi i utilitzar com a carta de presentació al saltar al món laboral. A partir d'un guió lògic es vol crear combinant disseny i programació, una interfície amb una distribució del contingut òptima. El resultat ha sigut una aplicació de disseny creatiu que respon a les necessitats funcionals requerides.

Resumen

Este proyecto tiene como objetivo la producción de una aplicación informática con contenido audiovisual, que permita comunicar la información deseada mediante recursos interactivos entre programa y usuario. Este producto digital se llama Portafolio Digital Multimedia. Una aplicación donde plasmar el Trabajo propio y utilizar como carta de presentación al saltar al mundo laboral. A partir de un guión lógico se quiere crear combinando diseño y programación, una interfaz con una distribución del contenido óptima. El resultado ha sido una aplicación de diseño creativo que responde a las necesidades funcionales requeridas.

Abstract

This project aims at producing a computer program with audiovisual content, to communicate the desired information through interactive resources between program and user. This product is called Digital Multimedia Portfolio. An application which reflects the own work and serves as a cover letter to jump to the labor market. Using a logical script we want to create combining design and programming, an interface with an optimal distribution of the content. The result has been an application with creative design that replies the functional need required.

Índex.

Índex de figures.....	III
Índex de taules.....	V
Glossari de termes.	VII
1. Objectius.....	1
1.1. Propòsit.	1
1.2. Finalitat.	1
1.3. Objecte.	1
1.4. Abast.	1
1.5. Objectius personals	2
2. Informacions sobre la realització del projecte/treball.	3
2.1. Motivació del projecte.	4
2.2. Justificació del projecte.	4
2.3. Pla de treball del projecte.....	4
2.3.1. Fase 1. Recopilació/Anàlisi de la informació i cerca d'inspiració.	4
2.3.2. Fase 2. Disseny de la interfície i disseny interactiu.	5
2.3.3. Fase 2. Programació.	5
2.3.4. Fase 3. Proves. Presa de mesures.	5
2.3.5. Fase 4. Redacció de la documentació.....	5
3. Preproducció.....	7
3.1. Fitxa tècnica.	7
3.2. Anàlisi de requeriments.	8
3.3. Disseny interactiu.	11
3.3.1. Disseny de la informació.....	11
3.3.2. Disseny de la navegació.	13
3.3.3. Interacció.....	15
3.4. Disseny audiovisual. Anàlisi de mèdies.	16
3.5. Guió lògic final.	18
3.5.1. Graf exhaustiu.	18
3.5.1. Fulls de configuració.....	19
3.5.1. Disseny i descripció de pantalles.....	20

4. Producció.....	27
4.1. Eines usades.....	27
4.2. Animacions.....	29
4.3. Esquema de la programació.....	30
4.3.1. Escena <i>intro</i>	30
4.3.1.Pre-càrrega.....	45
4.3.2. Menú principal i contingut.....	46
4.3.3. Posicionament i mode <i>fullscreen</i>	64
4.3.4. Sons.....	67
5. Postproducció.....	69
5.1. Publicació del portafoli.....	69
5.2. Proves d'usabilitat.....	71
5.3. Problemes i solucions.....	72
6. Impacte mediambiental.....	73
7. Propietat Intel·lectual.....	75
8. Conclusions i ampliacions.....	77
9. Referències.....	79

Índex de figures.

Fig 1. Estructura jerarquia bàsica d'un hipertext a dos nivells.....	10
Fig 2. Graf general de navegació de la <i>intro</i>	13
Fig 3. Graf general de navegació del portafoli.....	13
Fig 4. Esquema de navegació per UC en l'escenari <i>intro</i>	14
Fig 5. Esquema de navegació per UC en l'escenari <i>portafoli</i>	14
Fig. 6. Pantalla Intro – Joc interactiu.....	21
Fig. 7. Pantalla menú inici.....	22
Fig. 8. Botó animat del menú principal.....	23
Fig. 9. Elements comuns de les diferents seccions.....	24
Fig. 10. UC5 “Perfil”.....	25
Fig. 11. UC6 “Web&Gràfics”.....	26
Fig. 12. UC8 “Contacte”.....	27
Fig.13 Etiquetes disposades en la línia del temps de l'escena <i>portafoli</i>	54
Fig. 14. *Full de transicions <i>Tweener</i>	57
Fig. 15. Etiqueta “seccions”.....	59
Fig.16. Interior del clip de pel·lícula “contingut”.....	62
Fig.17. Configuració del codi <i>html</i> a publicar.....	75
Fig. 18. Configuració de l'espai creat on penjar l'aplicació.....	76
Fig. 19. Colocació de l'aplicació mitjançant <i>Dreamweaver</i>	77

Índex de taules.

Taula 1. Distribució d'unitats de contingut en escenes <i>intro</i> i <i>portafoli</i>	10
Taula 2. Mitjans audiovisuals distribuïts per a cada unitat de contingut.....	17
Fig. 6. Graf exhaustiu del <i>portafoli</i>	18
Taula 3. Patrons de comportament d'elements estables.	19

Glossari de termes.

<i>.swf</i>	Format de pel·lícula <i>Flash</i> .
GMA	Graduat en Mitjans Audiovisual.
CV	Currículum Vitae.
<i>html</i>	Llenguatge de marcat d'hipertext, llenguatge predominant en elaboració web.
css	Fulles d'estil en cascada: llenguatge usat per a definir la presentació d'un document estructurat en <i>html</i> .
<i>javascript</i>	Llenguatge de programació interpretat dirigit a objectes.
PC	<i>Personal computer</i> .
<i>.pdf</i>	Format de document portàtil.
UC	Unitat de contingut.
<i>.flv/.f4v</i>	Formats contenidors propietaris usats per transmetre vídeo per internet mitjançant <i>Adobe Media Player</i> .
OOP	Programació orientada a objectes.
<i>skip</i>	Terme àmpliament utilitzat en introduccions web que significa "salt" i actua com a salt de pantalla.
<i>Pepfolio</i>	Terme utilitzat per a referir-se al <i>portafoli</i> personal fent referència al candidat.
<i>Link</i>	Terme que vol dir "vinclé" i fa referència a accions de navegació web.
<i>Plugin</i>	Mòdul de hardware o software que afegeix una característica o serveix específic a un sistema més gran.
<i>.fla</i>	Format d'arxius <i>Flash</i> .
<i>Alpha</i>	Valor de transparència dels símbols en <i>Flash</i> .

Fullscreen Pantalla completa.

1. Objectius.

1.1. Propòsit.

Produir una aplicació digital multimèdia i interactiva amb material audiovisual per tal d'exposar de manera professional contingut i informació sobre el meu treball durant aquests anys d'estudi i iniciació al món audiovisual.

1.2. Finalitat.

Disposar d'una aplicació d'alta qualitat, útil i essencial per a presentar-se de cara al món laboral; i aconseguir un reconeixement per tal d'obrir-se portes i poder oferir el producte en el mercat.

1.3. Objecte.

Aplicació informàtica amb contingut audiovisual i multimèdia disponible en la web o en format portable (.swf) .

1.4. Abast.

S'especifiquen els documents i programes informàtics utilitzats en el projecte, així com tots els *scripts* necessaris per dur a terme l'aplicació. Es donarà a conèixer el procés de creació de l'aplicació detallant cada part de la preproducció i producció.

1.5. Objectius personals.

Un dels principals motius pels quals es va emprendre aquest projecte va ser la possibilitat d'ampliar els propis coneixements personals i inclús d'aprendre conceptes nous en cada una de les fases de la producció de l'aplicació.

En gran mesura, els coneixements adquirits serien de programació amb llenguatge *Actionscript 3* amb el programa *Flash*; realitzar una pel·lícula animada des de zero és una bona oportunitat per obrir-se en aquesta branca del multimèdia.

D'altra banda haurien augmentat els meus coneixements en treball de disseny vectorial i maquetació d'interfície gràcies al ampli nombre de gràfics i animacions que conté el projecte. Un repte que consisteix en realitzar un treball que requerirà reflexió, creativitat i lògica.

A tot això cal sumar-hi el més important, que és el fet de poder gaudir de les diferents eines que ens presenta la tecnologia d'avui en dia. Integrar-se a unes eines de les quals estan innovant amb rapidesa i que cada vegada són més amplies.

2. Informacions sobre la realització del projecte.

2.1. Motivació del projecte.

El projecte neix de la idea sobre la possibilitat de poder ajuntar dues branques generals en el món de la tecnologia digital: l'audiovisual i multimèdia. Un espai digital interactiu on agrupar el màxim d'elements de forma vinculada entre sí per a una exposició còmoda i rica visualment parlant. Crear d'aquesta manera una eina de comunicació útil i essencial aprofitant el màxim de vessants que avarca el món audiovisual/multimèdia.

És un projecte amb la motivació i el desig de realitzar una producció amb la qual exposar i ampliar tots els conceptes sobre la temàtica alhora de gaudir-hi produint-ho. Aconseguir un treball que pugui ser reconegut en el seu àmbit i que pugui donar un servei a nivell individual: diferents sortides en els aspectes audiovisual/multimèdia (disseny gràfic, animació, editor de vídeo, etc.). O un servei a nivell de mercat: produccions digitals d'alta qualitat dirigida a empreses, autònoms o clients privats.

Aquestes aplicacions multimèdia contenen tot tipus de contingut audiovisual, fet per el qual hi brinda aquest atractiu de més, al no tractar-se únicament d'aplicacions contenidores de només text i botons per exemple.

L'ambició de realitzar espais digitals multimèdia és el fet de posar en pràctica aquests anys d'estudi en el GMA i així tenir la oportunitat d'ampliar els coneixements en quan a programació, disseny i interactivitat. Una forma d'exposar el treball realitzat amb entusiasme i dedicació.

És així que amb aquest projecte es materialitza tota aquesta idea creant una aplicació multimèdia interactiva tipus *portafoli* anomenada *Pepfolio*.

2.2. Justificació del projecte.

El món de l'audiovisual cada dia i des de fa uns pocs anys, avança a pas de gegant. És una branca dins del món laboral de la qual s'hi podrien donar expectatives positives si en quan a innovació parléssim. Les noves tecnologies comencen a ser el "pa de cada dia" dintre de la nostra societat, i de manera ràpida però hàbil, aquestes ens van introduint canvis en la nostra vida. Una mostra d'això són les xarxes socials, existents des de fa bastants temps, però genèricament massives en els últims anys. El format digital és des de fa uns anys el sistema de comunicació predominant avui en dia i s'ha d'aprofitar.

És pensat en això que es va decidir el fet de crear un espai vistós i alhora elegant per a presentar el treball, en aquest cas personal. Un espai creat en format digital, el qual es pugui portar a sobre a tots moments o be deixar-lo publicat a la web.

La pretensió d'aquest projecte és la de poder disposar d'una eina necessària de nivell professional a nivell individual enfocada a aspiracions futures dins del món laboral.

En conclusió es pot dir que aquest projecte està pensat per a obrir-se portes al món laboral un cop acabat els estudis, i si més no per a ampliar conceptes dins d'aquest món. S'ha de dir que també hi té una mirada de futur cap al mercat on cada dia hi ha més persones i empreses que necessiten una carta de presentació digital rica en contingut i imatge.

2.3. Pla de treball del projecte.

El pla de treball del projecte es divideix en quatre fases principals pera la seva realització.

2.3.1. Fase 1. Recopilació d'informació i cerca d'inspiració.

Quan es decideix realitzar una aplicació multimèdia interactiva com a eina de presentació, cal primer pensar bé quina és la informació més convenient a exposar i seguidament fer un esforç creatiu per tal d'exposar aquesta informació de la manera més adequada. Per això, cal fer una cerca a fons per tal d'informar-se el màxim possible abans de començar a crear. També és primordial en un projecte d'aquestes característiques l'observació i l'interès per el material audiovisual i multimèdia pioner en l'època actual.

2.3.2. Fase 2. Disseny de la interfície i disseny interactiu.

En aquesta fase la creativitat pren un rol important. Es decideix quina forma obté el projecte i quin grau d'interactivitat se li aplica. Cal vincular tot el material audiovisual per tal de trobar l'harmonia desitjada en l'aplicació.

2.3.3. Fase 3. Programació.

En contrast a la fase 2, la fase 3 requereix de molta lògica i matemàtica. La programació ens permet crear les accions desitjades entre gràfics.

2.3.4. Fase 4. Proves. Presa de mesures.

Un cop creada l'aplicació s'haurà de respondre als seus *feedbacks* per tal de millorar-la i resoldre els problemes existents. Cal testejar el projecte fins aconseguir el desitjat.

2.3.5. Fase 5. Redacció de la documentació.

La redacció de la memòria és l'última fase per tal de deixar en escrit cadascuna de les propietats del nostre projecte.

3. Preproducció.

3.1. Fitxa tècnica.

- **Nom Producte:** Portafoli Digital Multimèdia CV.
- **Idea:** Produir una aplicació informàtica d'aspecte multimèdia interactiu per a exposar informació i contingut sobre una persona, col·lectiu o empresa.
- **Problema/Necessitat:** Un cop es decideix obrir-se portes al món laboral, quelcom essencial per algú dedicat a l'audiovisual és un bon *portafoli* digital d'alta qualitat per fer servir com a carta de presentació on poder exposar els treballs tant com les seves qualitats. Tant per algú a nivell individual com per una empresa on la imatge i qualitat de presentació pot ser determinant.
- **Objectius del producte:** Principalment aquesta producció s'ha realitzat mitjançant unes pautes d'un projecte de final de carrera, val a dir llavors que l'objectiu principal és el de formar a l'estudiant el qual ha realitzat un seguit de processos utilitzant els conceptes i les habilitats adquirides i s'ha de deixar que demostrï que es just mereixedor del títol a optar. També està marcat com a objectiu obrir-se portes a un món laboral.
- **Target principal:** Una aplicació digital multimèdia d'aquest tipus té un *target* molt ampli. A nivell individual acostumaria a ser algú ficat en l'entorn audiovisual per qüestions de contingut. A nivell col·lectiu o empresarial, el públic objectiu passaria a ser l'interessat amb una bona imatge i presentació de nivell professional.
- **Plataforma:** PC.
- **Maquinari i programari per al desenvolupament:** *Adobe Flash, Adobe Illustrator, Adobe Photoshop, Adobe Media Encoder, Adobe Dreamweaver* i llenguatges: *actionscript3.0, css, html, javascript*.

3.2. Anàlisi de requeriments.

La producció d'un *portafoli* digital multimèdia requereix la creació d'una aplicació informàtica capacitada per a suportar una certa quantitat d'arxius audiovisuals. Es tracta d'una aplicació dinàmica amb continguts multimèdia, vinculats entre sí. La idea es aconseguir una aplicació de nivell professional mesclant interactivitat amb animació, de manera que es percebi una ambientació i visió creativa.

Per analitzar el grau d'interactivitat de l'aplicació s'ha fet suport amb les 4 dimensions que proposa Toni Matas:

- Aleatorietat vs. Seqüencialitat
- Estructuració vs. Desestructuració
- Atemporalitat vs. Temporalitat
- Interactivitat vs. Passivitat

En el nostre projecte es pot dir que la nostra agulla gradual es declina cap a l'esquerra dels conceptes mostrats a la llista de forma accentuada.

Així és que es busca una aplicació amb una clara organització de la informació amb un graf de navegació ben adaptat per tal de donar una bona usabilitat i navegabilitat a l'usuari, evitant confondre'l. L'usuari en tot moment sabrà en quin lloc es troba dins de l'aplicació i tindrà el control per poder navegar amb aleatorietat sense problemes.

Llavors en el *portafoli* digital els continguts s'han dividit en dos grans escenes: la *intro* i el *portafoli*. A partir d'aquí, es definirà cada escena per separat.

A la *intro* els continguts s'han disposat seqüencialment al començar la pel·lícula transformant-se seguidament amb un joc interactiu, amb la característica de tenir la possibilitat de saltar al menú de selecció si així és vol. Un cop es perd al joc interactiu s'estableix un menú d'on pots triar tornar a jugar o entrar al *portafoli*. Això fa que la *intro* no sigui totalment seqüencial.

Així l'escena *intro* disposarà del següent contingut:

- **Intro:** el missatge de benvinguda en format text animat que dura pocs segons.
- **Joc interactiu:** petita aplicació amb forma de joc tipus *shooter* vertical.
- **Menú de selecció:** menú on es podrà triar mitjançant botons entrar al *portafoli* o tornar a jugar al joc.

Un cop a l'escena del *portafoli*, els continguts estan disposats en 4 apartats principals:

- **Perfil:** on s'hi troba tot el relacionat amb l'expedient del candidat amb una opció per a descarregar del mateix en format *.pdf*.
- **Web&Gràfics:** espai dedicat a visualitzar les plantilles web i gràfics realitzats pel candidat.
- **Vídeo&Animacions:** espai amb reproductor de vídeo per poder visualitzar les animacions realitzades pel candidat.
- **Contacte:** formulari per a ficar-se amb contacte amb el candidat.

A part dels 4 apartats principals, l'escena del *portafoli* optarà amb un enllaç addicional al *blog* personal del candidat, enllaç a les xarxes socials, opció pantalla completa i reproductor de música.

La pel·lícula està formada per diferents unitats de contingut que apareixen de forma estructurada en l'aplicació. A continuació es mostren les unitats de contingut d'aquest *portafoli* digital multimèdia per al projecte *Pepfolio*: (Les unitats variaran segons cada projecte diferent)

- UC1: *Intro*
- UC2: Joc interactiu
- UC3: Menú de selecció
- UC4: Escena *portafoli* (menú inici)
- UC5: Perfil
- UC6: Web&Gràfics
- UC7: Vídeo&Animació
- UC8: Formulari Contacte

- UCExt: Aquí formen part tots els continguts exteriors com enllaços a xarxes socials o a *blog* personal.

Les unitats de continguts estan distribuïdes entre les dos escenes de la següent manera:

<i>Intro</i>	<i>Portafoli</i>
UC1 – UC2 – UC3	UC4 – UC5 – UC6 – UC7 – UC8 - UCExt

Taula 1. Distribució d'unitats de contingut en escenes *intro* i *portafoli*.

Un anàlisi de les quatre dimensions de Toni Matas per cadascuna de les unitats de contingut ajuda a definir la nostra aplicació

- **UC1 – Intro.** Es tracta d'una animació seqüencial que passa a ser aleatòria i interactiva gràcies al botó *skip*. L'estructura es clara i concisa, fent entendre el missatge comunicat.
- **UC2 – Joc Interactiu.** Aquesta UC és totalment una aplicació interactiva. Conté una seqüencialitat que va creant objectes indefinidament, per tant és atemporal.
- **UC3 – Menú de selecció.** Aquí l'usuari es troba amb una sèrie de botons ben distribuïts per a triar quan ell vulgui i amb el temps que vulgui una opció (aleatorietat, atemporalitat, estructuració i interacció).
- **UC4 – Menú inci.** A dins del portafoli un cop al menú inici es podrà triar aleatòriament el camí a seguir en quant a la navegació. L'usuari es trobarà davant d'una estructura artística però formal a la vegada.
- **UC5 – UC6 - UC7 – UC8.** Totes aquestes unitats de contingut disposen de les mateixes dimensions d'interactivitat i estructuració. Es troben amb un camí seqüencial per a cadascuna a l'hora de tornar cap al menú inici per tal de canviar de UC entre si.
- **UCExt.** Aquestes unitats de contingut es disposen a l'escenari en forma de botó interactiu pel que fa als enllaços externs. El reproductor de música es presenta en forma de clip de pel·lícula però es tracta també d'un element 100% interactiu.

Cal remarcar que en aquest projecte es crearà un *portafoli* digital multimèdia i per tant el disseny d'interfície s'adequarà a les necessitats del candidat; volent dir que cada projecte del mateix àmbit emprés en el futur haurà de tenir el seu propi anàlisis de requeriments.

3.3. Disseny interactiu.

3.3.1. Disseny de la informació.

L'organització de la informació pot ser determinant per a l'usuari per a decidir si estarà disposat a gastar el seu temps navegant per l'espai multimèdia o si per altra banda, perdrà la paciència i abandonarà el lloc per no entendre o no trobar el que busca. És per això que cal estructurar i clarificar el contingut per tal de permetre a l'usuari trobar una via de navegació cap a la comprensió i manipulació de la informació.

El portafoli es presenta inicialment amb la primera escena la qual tracta d'una animació d'uns segons informant d'on s'està entrant seguit del joc interactiu tipus "*shooter*". En tot moment l'usuari està informat del que pot fer; anar directament al menú de selecció clicant el botó *skip* que es posiciona centrat a la part superior de la pantalla de joc; o bé esperar a perdre la partida per triar si vol tornar a jugar o entrar a l'aplicació. Un altre element informatiu que trobem a la pantalla *intro* és la puntuació acumulada aconseguida per l'usuari mentre està jugant.

Al entrar al *portafoli*, l'usuari pot començar a descobrir els apartats i els continguts que conté l'aplicació. S'ha de dir que l'usuari estarà en tot moment informat de l'apartat en que es troba mitjançant els títols de capçalera que estaran establerts en cada un d'ells.

La distribució de la informació és simple i concisa, de forma jeràrquica els continguts s'organitzaran en seccions principals i secundaries.

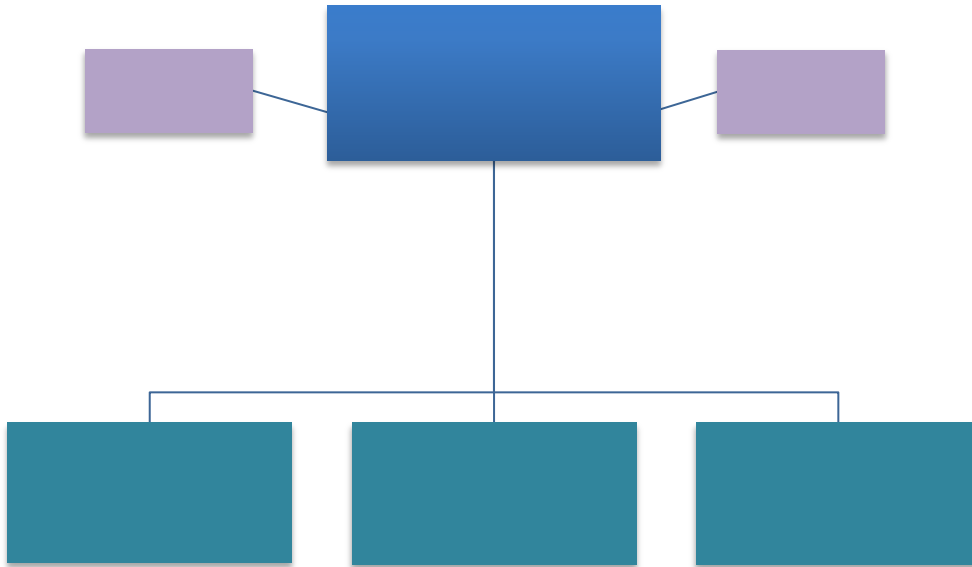


Fig 1. Estructura jeràrquica bàsica d'un hipertext a dos nivells

La posició que ocupa cada element a la pel·lícula és important. Ja que no a tota la informació se li dona la mateixa importància per a comunicar bé el que es vol transmetre a l'usuari, al determinar la jerarquia de la informació ens trobem amb contingut principal i contingut secundari, així com apartats principals i apartats secundaris.

En quant a la maquetació dels elements en la pàgina es basa amb el model occidental, llegint d'esquerra a dreta i de dalt cap a baix. Així com es presenta la informació principal en la part superior (títols) per a que l'usuari tingui clar sempre on es troba. Si per exemple l'usuari es troba dins d'una secció anomenada "Vídeo/Animació", es situa un títol situat a la part superior comunicant a quin apartat es troba per a una millor comprensió.

Els apartats es troben centrats en forma de botons animats en la pàgina principal de la pel·lícula. Un cop dins de l'apartat, un botó animat a la part esquerra superior ens guia al menú principal. Els botons, que són elements principals per a la guia de la navegació, es presenten en un format idoni en quant a mida i color per tal de cridar l'atenció de l'usuari i així fer-li més eficaç la experiència.

Amb aquesta distribució de la informació, l'usuari és capaç d'entendre els continguts de manera simple i eficaç mentre busca la informació desitjada.

Aquesta distribució del contingut no només és útil per a l'usuari sinó que també ajuda al creador de l'aplicació alhora d'actualitzar l'espai de manera fàcil i ràpida.

3.3.2. Disseny de la navegació.

Graf General

Al iniciar-se l'aplicació visualitzem la *intro* on es pot escollir jugar una estona al joc de tipus *shooter* o entrar directament al *portafoli* escollint l'idioma desitjat. Aquesta interacció es creada en forma de botons.

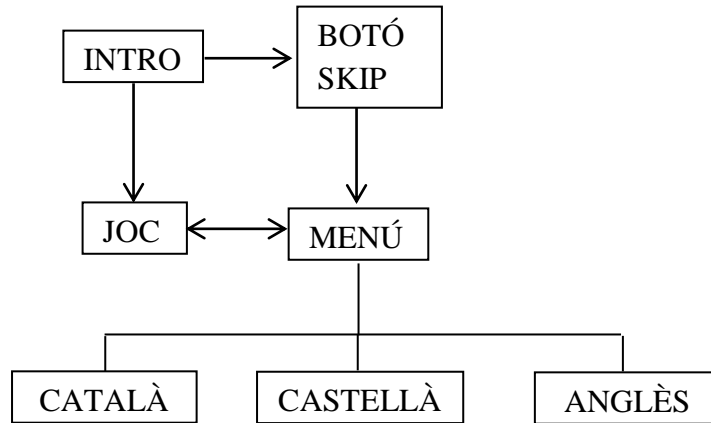


Fig 2. Graf general de navegació de la *intro*.

A continuació es presenta el graf general de navegació de forma esquemàtica un cop a dins del *portafoli*.

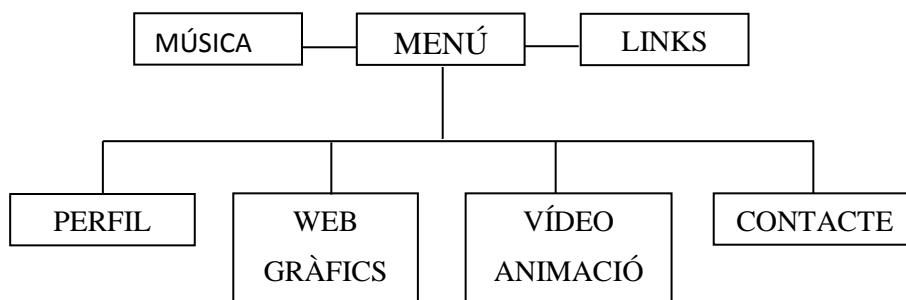


Fig 3. Graf general de navegació del portafoli.

Navegació

A l'escena *intro* la navegació és aleatòria, en tot moment l'usuari pot triar on vol anar i què vol fer. Si de la *intro* vol passar al menú de selecció disposa del botó *skip*, si per contra vol jugar al joc de tipus *shooter* serà capaç de fer-ho des de la *intro* o bé des del menú interactiu. L'esquema de navegació té aquesta forma:

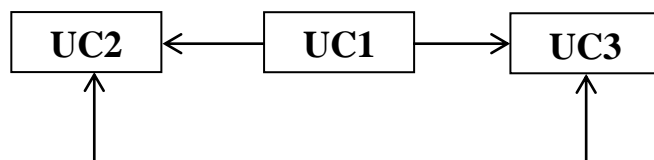


Fig 4. Esquema de navegació per UC en l'escenari *intro*.

Al entrar al *portafoli* (UC3), l'usuari visualitza el menú d'inici de l'aplicació, unitat de contingut que marca la referència durant tota l'experiència. L'usuari, al poder triar quina àrea de contingut visualitzar primer es troba en una navegació aleatòria. Però la seqüència de passar d'una UC (continguda al menú principal) a una altra és de forma lineal. És a dir, des de dins d'una UC del menú principal s'ha d'engegar el botó de retorn al menú principal per poder tornar a realitzar la tria aleatòria de contingut.

Per tant l'escena del *portafoli* té una navegació semi-aleatòria. A continuació es mostra l'esquema de navegació referent:

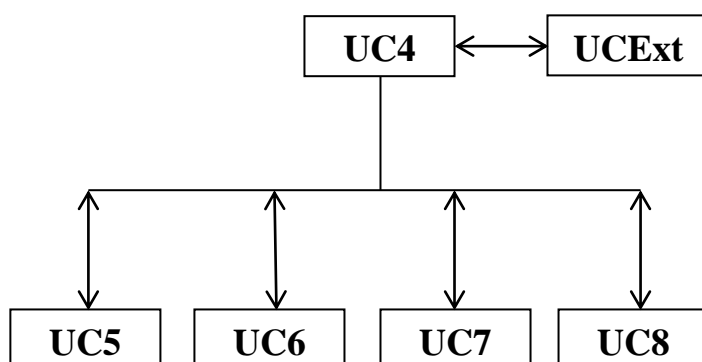


Fig 5. Esquema de navegació per UC en l'escenari *portafoli*.

3.3.3. Interacció.

El tipus de comunicació entre l'aplicació i l'usuari, és a dir, l'orientació, és de manera guiada i estructurada en tot moment fent servir menús i botons. Cada botó de la pel·lícula té un aspecte similar amb un disseny òptim per a que l'usuari intueixi el seu camí de forma clara. Aquests botons estan presents en cada pantalla de la pel·lícula, mostrant a partir d'una petita animació el camí que porta.

L'aplicació en general no conté temporalitat a excepció de l'UC exterior referent al reproductor de música. Encara que podem dir que el reproductor té una dimensió de temps seqüencial, sempre es pot interactuar amb aquest de manera que si es vol canviar de tema musical o es vol silenciar-lo per exemple, l'usuari n'és lliure. L'altre petit tram seqüencial de la pel·lícula el trobem a la *intro* on l'usuari es lliure també de visualitzar els segons de presentació o passar directament a l'escena del portafoli.

El **grau d'interactivitat** és immediat ja que en tots els casos es basa en una interacció causa/efecte del qual l'usuari n'és conscient en tot moment, sense haver de perdre's.

Una vegada que l'usuari entri a la nostra aplicació, serà capaç de triar el que vol fer en tot moment de manera que si decideix veure la *intro* i després jugar-hi una estona per en acabat entrar al *portafoli* ho podrà fer. O si decideix entrar-hi directament al *portafoli* també estarà en la llibertat de fer-ho. Un cop dins del *portafoli* l'usuari descobrirà i intuirà visualment la navegació entre els continguts per a poder efectuar una navegació aleatòria. Això vol dir que oferirem una aplicació amb **accés aleatori** on l'usuari tindrà el **control** tota l'estona.

A continuació es descriuen els elements i accions interactives que contenen cada unitat de contingut de l'aplicació:

- **UC1 – Intro:** l'element interactiu en aquesta UC és el botó *skip* per tal de saltar al menú de selecció.
- **UC2 – Joc Interactiu:** tots els elements d'aquesta UC són interactius, és pot parlar d'una interactivitat total.
- **UC3 – Menú de selecció:** els botons del menú aquí, són els responsables d'interactuar amb l'usuari i la seva decisió.

- **UC4 – Menú inici:** el menú principal del *portafoli* conté vinculació amb totes les unitats de contingut disposades en aquesta escena. Mitjançant els seus botons, la manera de comunicar la informació és clara i organitzada.
- **UC5 – Perfil:** l'apartat del perfil del candidat disposa d'un *link* per a la visualització i possibilitat de descarrega del CV d'aquest.
- **UC6 – Web&Gràfics:** la interactivitat en aquest apartat sorgeix quan l'usuari clica les imatges disponibles, i aquestes mitjançant una interpolació de moviment adapten una mida superior.
- **UC7 – Vídeo&Animació:** un reproductor de vídeo és l'element interactiu en aquesta UC.
- **UC8 – Contacte:** Un formulari amb requadres de text dinàmics per a omplir permet ficar-se en contacte amb el candidat via correu electrònic.
- **UCExt:** la interactivitat en aquestes unitats de contingut són de tipus botó amb *links* a enllaços exteriors. També existeix un botó que transforma l'aplicació a mode pantalla completa. Pel que fa al reproductor de música, s'hi disposa de forma manejable per a un total control de les funcions típiques (*play, pause, mute...*).

3.4. Disseny audiovisual. Anàlisi de mèdies.

Com a tota pel·lícula interactiva, el nostre projecte conté diferents mitjans audiovisuals per tal de presentar el contingut. De forma que s'utilitza més d'un mitja audiovisual com a eina de comunicació a la vegada. El *portafoli* digital multimèdia conté: text, imatges vectorials, imatges *bitmap*, vídeo, animacions i so.

- **Text.** El mitjà essencial per tal de comunicar de manera literal. Es troba a cada UC en major o menor magnitud per tal de facilitar la comprensió de l'usuari.
- **Imatge vectorial.** Aquest element es disposa a cada part de la pel·lícula. La interfície de l'aplicació està creada a partir d'imatges vectorials així com moltes de les animacions.
- **Imatge *bitmap*.** Les imatges de tipus *bitmap* es situen a l'apartat Web&Gràfics per ensenyar les feines realitzades.
- **Imatge dinàmica.** Es presenten tots els vídeos i animacions realitzades pel candidat en la secció Vídeo&Animació.

- **Animacions.** Les animacions establerts en l'aplicació seran conjunts d'imatges vectorials transformades en interpolacions de moviment gràcies al programa Flash i el llenguatge *Actionscript 3.0*.
- **So.** El so estarà present en tot moment, encara que l'usuari podrà triar si vol àudio o no. L'àudio estarà exposat en la UC del reproductor de música i en diferents elements de la *intro* igual que en interactuar en la escena del *portafoli*; per exemple al arrossegar el ratolí sobre dels botons d'interacció.

Un altre aspecte important de l'anàlisi de mèdies és la **volatilitat** de l'aplicació. Bàsicament la nostra pel·lícula la podem exposar de forma dinàmica a la web, ja que un cop publicada a internet podrà ser actualitzada sense problema. Per a la versió portàtil en format *.swf*, la podem transportar amb cd, *pendrive* etc. De totes maneres en aquesta versió serà estàtica i aquesta no es podrà actualitzar. Una solució és ficar un enllaç web en la versió portàtil per tal de visitar la versió online actualitzada.

La següent taula mostra com estan distribuïts els diferents mitjans audiovisuals de l'aplicació a les unitats de contingut (UC).

Unitats de contingut	Text	Imatge vectorial	Imatge <i>bitmap</i>	Imatge dinàmica	Animació	So
UC1						
UC2						
UC3						
UC4						
UC5						
UC6						
UC7						
UC8						
UCExt						

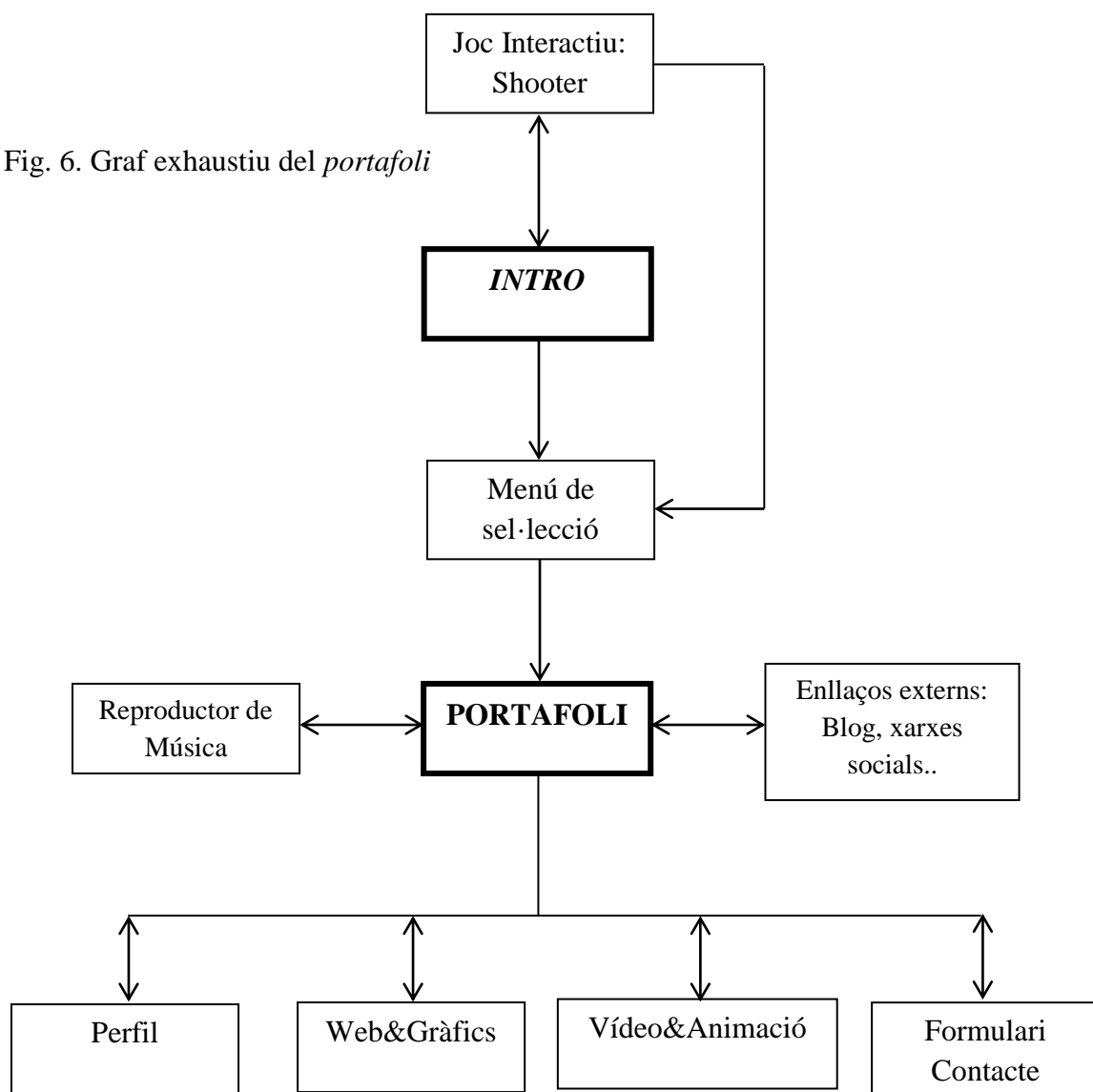
Taula 2. Mitjans audiovisuals distribuïts per a cada unitat de contingut

3.5. Guió lògic final.

Aquest punt és imprescindible en la fase de preproducció. Aquí és on es deixen indicades les especificacions per a que l'equip de producció sàpiga què fer i com ha de seguir la producció de l'aplicació. Es defineix el disseny interactiu de forma detallada incloent-hi el graf exhaustiu, el disseny de pantalles i estil, i la descripció de pantalles i unitats de contingut.

3.5.1. Graf exhaustiu.

Per a descriure amb detall una zona concreta del graf general i representar les seqüències i els possibles ramals de l'aplicació, realitzem el graf exhaustiu. Aquest graf visualitza el comportament dels elements a nivell d'escena o unitat de contingut.



3.5.2. Fulls de configuració.

Elements estables

Alguns elements són comuns a més d'una pantalla, s'anomenen **elements estables**. Es comporten de la mateixa manera i apareixen en els mateixos llocs encara que l'escena o unitat de contingut sigui diferent.

En la nostra aplicació hi ha elements estables en cada una de les dos escenes que tenim, a la *intro* i al portafoli. En la *Intro* l'únic element estable que s'hi troba és el marge que exerceix de límit del joc interactiu de tipus *shooter*. En el *portafoli* hi ha més d'un element estable i són els següents: el reproductor de música, el botó de pantalla completa, el logo del *portafoli* i els enllaços al *blog* personal i xarxes socials.

Patrons de comportament

En aquest punt es descriu i s'especifica el que succeeix i quins són els esdeveniments que es donen en els elements estables de l'aplicació. Per a la tasca s'utilitza una taula per a que quedi clarificat.

Element Estable	Escena	Esdeveniment	Descripció
Marge <i>Shooter</i>	1 (<i>Intro</i>)	Paret límit.	Marge que exerceix de límit per al joc interactiu <i>Shooter</i>
Reproductor Música	2 (Port.)	Reproduir música	Interfície amb control absolut de l'àudio
Logo portafoli	2 (Port.)	Enriquidor visual	Logotip en format vectorial
Botó <i>fullscreen</i>	2 (Port.)	Visualitzador a pantalla completa	Botó amb petita animació per tal de passar al mode <i>fullscreen</i>
Enllaços externs	2 (Port.)	Enviar usuari a espais externs del candidat	Botons amb petites animacions per a vincular el portafoli amb espais externs del candidat

Taula 3. Patrons de comportament d'elements estables.

3.5.3. Disseny i descripció de pantalles.

El disseny i descripció de pantalles (unitats de contingut) és la part més important per aconseguir exposar els continguts de manera que s'aconsegueixi comunicar realment el que volem transmetre a l'usuari. Depenent del disseny de pantalles que es construeixin es transmet un missatge a l'usuari o un altre, es depenent d'aquest apartat que la pel·lícula pot ser recordada o no.

Al produir la interfície de l'aplicació multimèdia cal analitzar la millor manera de tractar les diferents respostes dels esdeveniments interactius. Per tant en aquest punt es dona especial importància als modes d'accés i a l'exposició de la informació i continguts.

Per al *portafoli* digital multimèdia hem optat per un disseny vistós, dinàmic i a la vegada sòlid. Això ve a dir que es necessitarà un treball gràfic artístic, una vinculació entre continguts amb animacions intercalades i una maquetació consistent. Queda clar que al tractar-se d'un *portafoli* personal s'intenta emfatitzar la vessant artística del disseny i queda més allunyat del disseny formal; el qual no vol pas dir que el disseny formal no hi estigui present. Encara que es tracti d'un *portafoli* personal, també es defineix com una carta de presentació digital, per tant desistirem de tot element que ultrapassi la informalitat per no crear pantalles que passin a transformar-se en vulgars.

Al entrar a la nostra aplicació, la primera pantalla que l'usuari visualitza és la seqüència animada de la *Intro* on mentre es crea la interfície del joc interactiu de tipus *shooter*, un seguit de text animat dona la benvinguda. Tot seguit, al acabar el missatge de benvinguda, l'usuari pot entendre que al aparèixer una nau pot començar a disparar estels i augmentar la seva puntuació disponible a visualitzar en un requadre de text dinàmic a la part inferior esquerra. En aquesta pantalla hi tenim situat centradament a la part superior del joc un botó (*skip*) per tal de passar al menú de la *Intro* que també apareix un cop ens eliminen al joc. Aquest menú consta d'un seguit de botons els quals ens envien a jugar un altre cop o ens inviten a entrar al *portafoli* podent elegir l'idioma desitjat. En aquesta pantalla, a sota del menú també visualitzem un recuadre de text dinàmic, on l'usuari podrà comprovar la seva puntuació final així aconseguint un efecte de continuïtat pel que fa referència al joc. S'ha de matisar en que es vol donar una ambientació més a la “vella escola” i per això es fan

servir tipografies i gràfics tractats de forma que donen aquesta sensació. Es situa al centre de la pantalla.

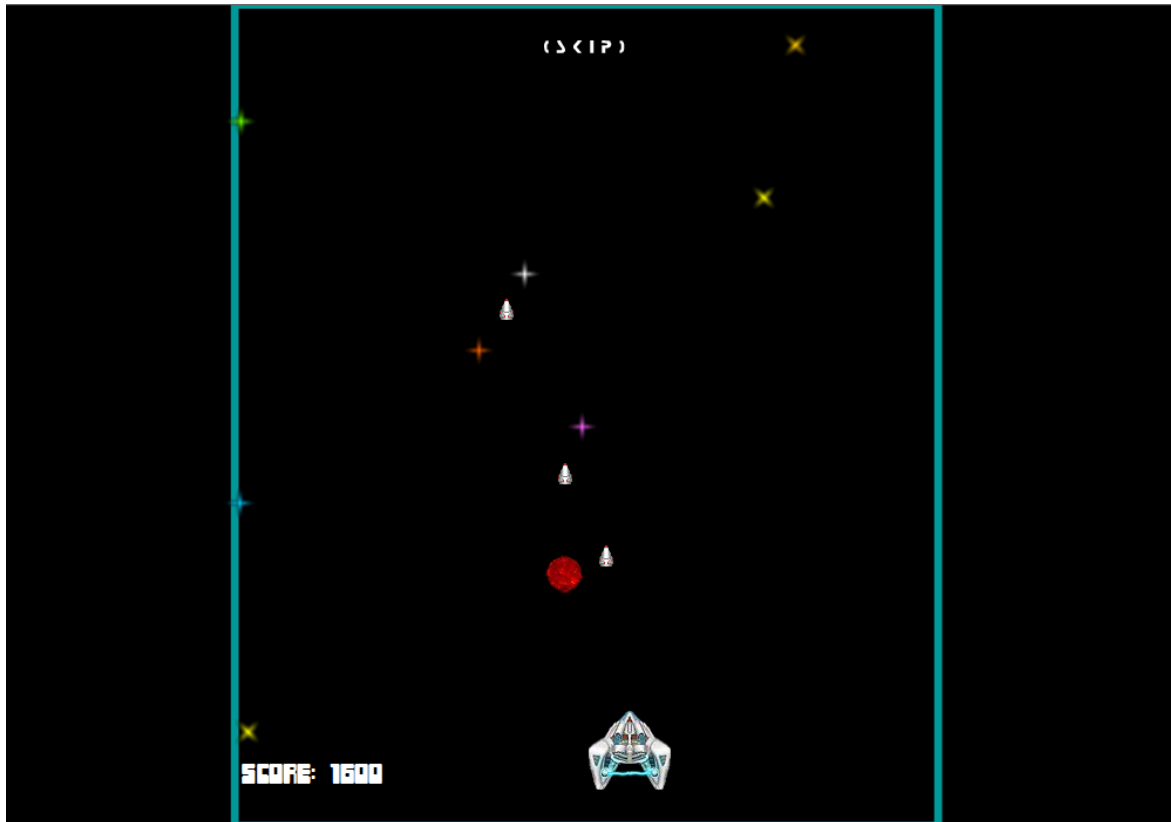


Fig.6. Pantalla Intro – Joc interactiu

Quan s'envia l'usuari cap al *portafoli*, aquest visualitza una escena addicional destinada a la pre-càrrega de la pel·lícula. Aquesta pantalla de mode estàndard té un fons pla i llis. Consta d'un parell de gràfics vectorials en forma de rectangle en el que un d'ells es va superposant mentre la càrrega de l'aplicació va avançant. S'hi estableix també un quadre de text dinàmic anunciant el tant per cent carregat fins al moment.

Un cop situats a la pantalla principal del *portafoli*, l'usuari percep un impacte visual del qual intuïtivament pot comprendre que està dins del *portafoli* gràcies al logotip situat a la capçalera; que es troba davant d'un menú d'inici gràcies als botons circulars animats al centre de la pantalla; que disposa de l'opció per escoltar diferent música o bé apagar-la gràcies al reproductor situat a la part inferior esquerra; que si vol, pot engegar el mode

pantalla completa amb el botó situat a la part inferior dreta; o si prefereix, pot visitar els enllaços a altres espais referents al candidat com el *blog* o les xarxes socials.

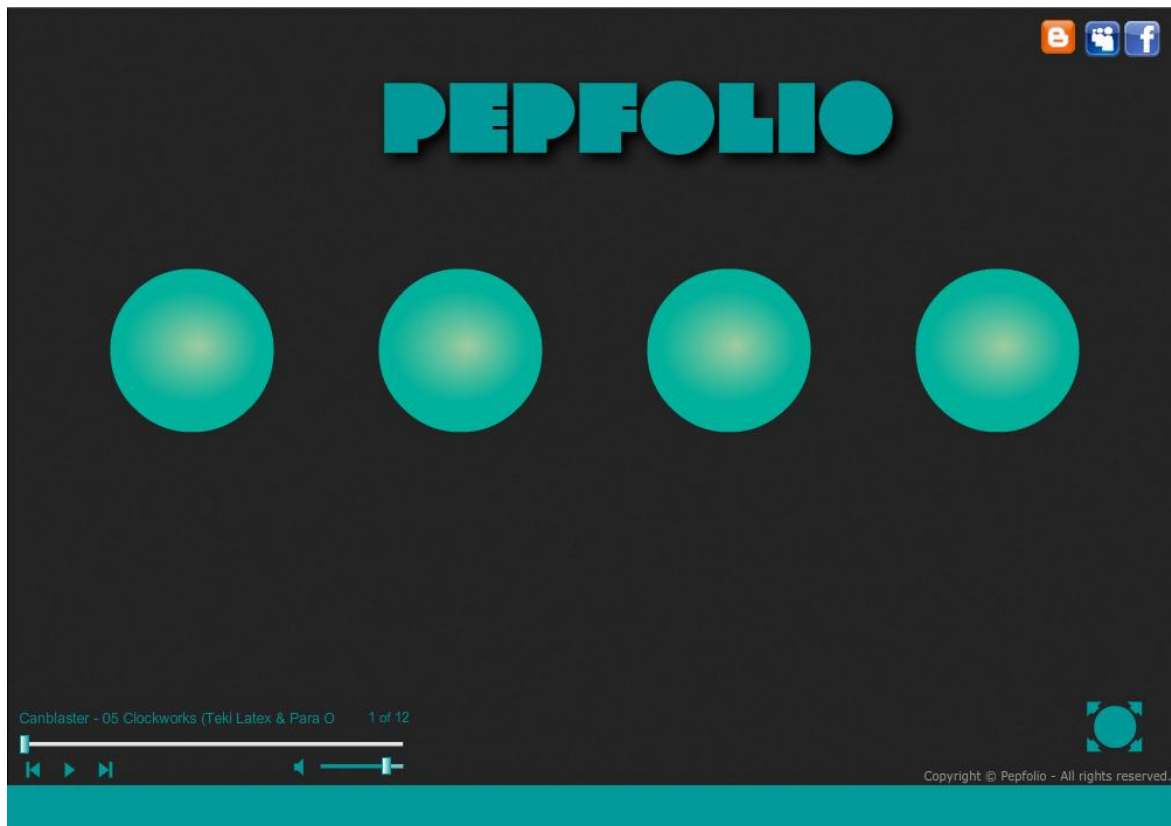


Fig.7. Pantalla menú inici

Descrita la pantalla principal, és moment de descriure les diferents UC establertes en el menú d'inici. Tenim un total de quatre diferents pantalles on accedir aleatòriament mitjançant els botons animats. Per a cada botó es reproduceix una petita animació en el moment d'arrossegar el ratolí per sobre d'aquest, amb un efecte tipus etiqueta que destapa el títol de la secció. A la part inferior dels botons s'hi visualitza una altra animació, aquesta vegada una animació diferent per a cada botó representant simbòlicament la secció a la qual ens envia.



Fig.8. Botó animat del menú principal

Un aspecte important que s'inclou per tal de donar dinamisme a la pel·lícula és la implantació d'animacions tipus *Tweener* (paquet de classes *Actionscript 3.0*). El que produeix aquesta implantació són les animacions en la transició que va des de que es pitja el botó del menú d'inici, a l'aparició de la pantalla de cada secció.

Les pantalles de les diferents seccions apareixen totes al mateix lloc, centrades a la pantalla. L'espai disponible per al contingut és el mateix per a cada secció. En cas de que l'espai de la pantalla sigui insuficient, es disposa d'un *scrollbar* a la part dreta de la informació. Mitjançant una mascara, el contingut pot adoptar moviment vertical al desplaçar l'*scrollbar*.

L'últim element comú que comparteixen les pantalles de les diferents seccions és el botó per retornar al menú d'inici. Es tracta del mateix botó amb efecte etiqueta però amb títol referent a "tornar al menú. Aquest es situa a la part superior esquerra del contingut de la secció.

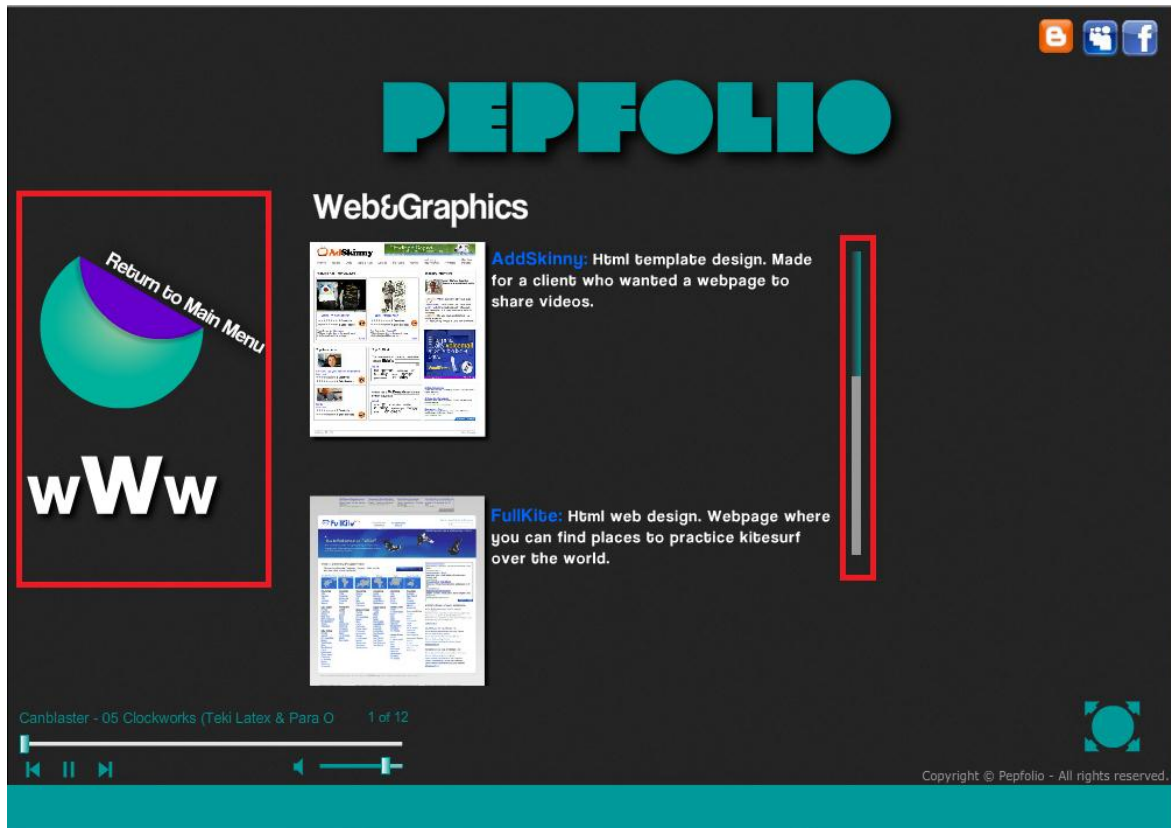


Fig. 9. Elements comuns de les diferents seccions.

El primer botó començant per l'esquerra envia a la UC5 del perfil de candidat, on mitjançant text ens informarà a partir d'una breu introducció de les seva història i expectatives. A la versió *online* del projecte estarà disponible una versió del CV del candidat en *.pdf* per descarregar. Encara que aquest fitxer requerirà de contrasenya per tal de poder-lo descarregar; bàsicament per tal de no publicar dades personals al web i així evitar qualsevol problema.



Fig.10. UC5 "Perfil"

La UC5 anomenada Web&Gràfics disposarà d'un recull d'imatges *bitmap* situades a la part dreta de la secció amb una breu explicació de cadascuna a la part esquerra d'aquesta. Aquestes imatges *bitmap* contindran treballs gràfics realitzats pel candidat, les quals podran ser ampliades al clicar-hi a sobre.



Fig.11. UC6 “Web&Gràfics”

La UC6 disposarà d'un reproductor de vídeo per a veure totes les creacions audiovisuals del candidat.

La quarta i última secció està dedicada a un formulari a omplir per tal de ficar-se amb contacte amb el candidat. Tracta d'un formulari estàndard on l'usuari ha d'omplir un seguit d'apartats en blanc com el nom, la seva adreça de correu electrònic, la raó, i els comentaris que vulgui afegir. Esta dissenyat per funcionar amb sistema *PHP*, per tant el servidor on hi publiquem l'aplicació haurà de suportar aquest llenguatge.



Fig. 12. UC8 “Contacte”

4. Producció.

4.1. Eines utilitzades

- Adobe Flash CS5:

Aquest programa conegut com a un dels pioners al fer referència a animació 2D i interacció amb l'usuari, ha sigut l'eina principal per a dur a terme el projecte. Es tracta d'una eina d'autor destinada a la producció de continguts interactius en forma d'estudi d'animació. Funciona en base una línia del temps on els fotogrames (*frames*) són els contenidors del contingut produït.



Tot el material creat a partir d'altres programes utilitzats (imatges vectorials, vídeo...) ha estat organitzat i tractat per a la seva interacció dins d' *Adobe Flash*.

Tota aplicació interactiva requereix de programació, i en el nostre projecte s'ha utilitzat el llenguatge *Actionscript 3.0* establert a *Adobe Flash*. *Actionscript 3.0* és la versió de llenguatge més expandida i més nova del moment. És un llenguatge de programació orientat a objectes (OOP) utilitzat especialment per a aplicacions web animades i animacions per a interactius. Això vol dir que l'aplicació no requereix d'un programa complet per a produir el nostre objectiu, sinó que és programació orientada al diferent contingut que sostinguin els fotogrames.

Adobe Flash CS5 ha sigut el programari més utilitzat en tot el procés de producció del projecte.

- Adobe Photoshop CS5:

Adobe Photoshop, programa pioner a nivell mundial per a tractament d'imatge tipus *bitmap* (fotografies) ha sigut l'eina principal utilitzada per a crear cert contingut de la interfície gràfica com per a tractar les imatges que conté l'aplic



El retoc d'imatge avui en dia és essencial per a una bona presentació d'un interactiu. Amb les noves funcions que va generant aquest programa així que avança el temps com els espais de treball multicapa, el tractament extensiu de tipografies, el control i retoc de color, llum i contrast; la inclusió d'elements vectorials, el tractament de textures i els efectes creatius, ha sigut de gran utilitat per a fer de la nostra aplicació un interactiu més atractiu.

- *Adobe Illustrator CS5:*

Si *Adobe Photoshop CS5* és el programa pioner per a tractament d'imatges tipus *bitmap*, *Adobe Illustrator* presenta la millor manipulació d'imatges vectorials. És un programa en forma de taller d'art per a treballar sobre un taulell de dibuix.



La gran versatilitat a l'hora de tractar imatges vectorials que ofereix ha sigut clau per a produir el que són els gràfics de l'aplicació de manera còmoda i professional. Aquesta eina d'autor també passa a ser essencial ja que tota aplicació interactiva sempre requereix d'imatge vectorial per tal de no perdre qualitat d'imatge al canviar de mida. Això significa que si la nostra aplicació es visualitza en diferents monitors amb resolucions diferents, no perd qualitat i sempre és veure com nosaltres volem i amb la màxima qualitat possible.

- *Adobe Dreamweaver CS5:*

Aquest programa és una aplicació en forma d'estudi (basada en al forma d'estudi d'*Adobe Flash*) que està destinada a la construcció i edició d'espais i aplicacions web basats en estàndards.

S'ha fet servir per tal de publicar la nostra aplicació *Flash* a internet i modificar alguns paràmetres en el codi *html* d'aquest.



- Adobe Media Encoder CS5:

Aquest és un altre programari d'autor que ha permès publicar els arxius de format imatge dinàmica. La principal funció del programa és la de convertir els vídeos que vulguem col·locar com a contingut de l'aplicació dins de *Flash* a un format compatible. El format en qüestió pot ser el *.flv* i *.f4v*, els quals són els formats de vídeo compatibles per a aplicacions interactives *Flash*.



4.2. Animacions.

Una part important de la producció són les animacions que conté la nostra aplicació. Aquestes animacions es produeixen amb el programa *Flash*.

Les animacions amb aquest programa s'han creat basant-nos amb interpolacions de moviment a la línia del temps. Així una vegada s'hagin realitzat els gràfics o text requerits, s'importaran al *Flash* i se'ls hi dona vida.

Seguidament es mostra un llistat de totes les animacions de l'aplicació contingudes en la línia del temps amb les seves referents definicions i característiques:

- **Creació de l'escenari del joc:** Aquesta és la primera animació que l'usuari visualitza al entrar a la nostra aplicació. És una simple interpolació de moviment de 50 fotogrames de durada, en la que una imatge gràfica que fa de fons és posiciona al centre de l'escenari de la pel·lícula. Aquesta imatge té uns marges que s'expandeixen a la mateixa vegada.
- **Text de benvinguda:** també s'ubica en l'escena *intro* i és una animació de text que va apareixen mentre dona un missatge de benvinguda. L'animació es creada amb una interpolació de moviment modificant el valor de la propietat *alpha*.
- **Botó del menú principal:** aquest botó està creat com a símbol de clip de pel·lícula. És el botó plantilla que és repeteix per a cadascuna de les seccions accessibles en el menú principal. Aquest botó ha estat creat a partir de varis objectes a la vegada, que amb

l'ajuda de les interpolacions de moviment i capes mascara aconsegueix produir un efecte d'etiqueta desenganxant-se (veure fig.X.).

- **Transició contingut:** les diferents pàgines de contingut de l'aplicació es disposen dins d'un clip de pel·lícula. A aquest clip de pel·lícula se li aplica una animació en la línia del temps per tal de crear la transició per aparèixer i desaparèixer de l'escenari.
- **Botó pantalla completa:** es tracta d'una petita animació dintre d'un clip de pel·lícula en la que quatre petites fletxes es mouen cap als quatre vèrtexs.

4.3. Esquema de la programació.

Un cop tots els continguts necessaris (gràfics, fotos, vídeos,...) s'han estat col·locat de la forma en que el guió lògic diu, és moment de passar a la programació. En aquest punt de la producció és on s'aconsegueix fer interactuar i vincular tots els objectes entre sí i així aconseguir un medi digital de comunicació amb el nostre usuari. Com ja hem dit anteriorment s'utilitza el llenguatge *Actionscript 3.0* establert en l' *Adobe Flash CS5*.

Ja que la programació *Actionscript* és orientada a objectes (OOP) s'anirà explicant per parts cada codi amb la seva funció sobre el contingut establert.

Comencem per la programació disposada en l'escena *Intro* on les accions interactives que s'hi troben són el botó anomenat *skip* que ens envia al menú de sel·lecció, el joc de tipus *shooter* vertical i el menú de sel·lecció. La *Intro* tracta d'una animació al llarg de la línia del temps on en l'últim fotograma s'acaba i s'hi disposa el primer codi de programació. En aquesta escena són el fotograma cinquanta-u i els dos últims fotogrames que disposen de codi, aquests últims estan l'un després de l'altre al final de l'animació en la línia del temps.

4.3.1. Escenari *intro*

Botó *skip*

Aquest botó apareix en meitat del missatge de benvinguda de la *Intro* i serveix per als usuaris que no vulguin veure la *Intro* i vulguin entrar directament al menú de sel·lecció per tal d'entrar al *portafoli*. El codi utilitzat per tal d'enviar l'usuari al menú de sel·lecció al prémer el botó *skip* és el següent:


```
// Crear un listener que escolti la instància "btskip"
btskip.addEventListener(MouseEvent.CLICK, fskip);

//Defineix la funció per a passar directament a l'etiqueta "lose" on s'hi disposa el menú de
sel·lecció
function fskip(event:MouseEvent):void{

gotoAndStop('lose');
}
```

Joc Interactiu

El joc tipus *shooter* vertical consta de 3 elements principals, la nau, els míssils i els enèmics. El primer a programar és la nau per tal de que es pugui moure per l'escenari. Llavors per a la nau amb nom d'instància "nau" la funció que s'ha utilitzat es descobreix en el codi següent:

```
//Aquests booleans serviran per veure kines tecles estan premudes
var leftDown:Boolean = false;
var upDown:Boolean = false;
var rightDown:Boolean = false;
var downDown:Boolean = false;

//Velocitat de la nau per tecla premuda
var velnau:int = 5;
```

```
//Funció per a moure la nau
nau.addListener(Event.ENTER_FRAME, funcionsEFrame);
function funcionsEFrame(event:Event):void{

//Compravarem el valor dels booleans i llavors mourem la nau
    if(leftDown == true){
        nau.x -= velnau;
    }
    if(upDown == true){
        nau.y -= velnau;
    }
    if(rightDown == true){
        nau.x += velnau;
    }
    if(downDown == true){
        nau.y += velnau;
    }

//Mantindrem la nau a dins dels marges del joc

    if(nau.x <= 200){
        nau.x += velnau;
    }
    if(nau.y <= 3){
        nau.y += velnau;
    }
    if(nau.x >= 800 - nau.width){
        nau.x -= velnau;
    }
    if(nau.y >= 702 - nau.height){
        nau.y -= velnau;
    }
}
```

```
//Aquest "listener" escoltarà les tecles sent premudes
stage.addEventListener(KeyboardEvent.KEY_DOWN,checkKeysDown);
function checkKeysDown(event:KeyboardEvent):void{

//Donant valors "true" als booleans mitjançant el codi WASD o les tecles de fletxes
    if(event.keyCode == 37 || event.keyCode == 65){
        leftDown = true;
    }
    if(event.keyCode == 38 || event.keyCode == 87){
        upDown = true;
    }
    if(event.keyCode == 39 || event.keyCode == 68){
        rightDown = true;
    }
    if(event.keyCode == 40 || event.keyCode == 83){
        downDown = true;
    }
}

//Aquest "listener" escoltarà les tecles sent deixades
stage.addEventListener(KeyboardEvent.KEY_UP, checkKeysUp);
function checkKeysUp(event:KeyboardEvent):void{

//Fent els booleans "false" basant-nos amb el codi de teclat
    if(event.keyCode == 37 || event.keyCode == 65){
        leftDown = false;
    }
    if(event.keyCode == 38 || event.keyCode == 87){
        upDown = false;
    }
    if(event.keyCode == 39 || event.keyCode == 68){
```

```

rightDown = false;
}
if(event.keyCode == 40 || event.keyCode == 83){
    downDown = false;
}

```

El següent element a programar són els míssils que volem que la nau dispari al prémer la tecla *space*. Per a aquesta acció s'han utilitzat les classes “*as*”. Les classes dintre del llenguatge *Actionscript* són codis externs a la pel·lícula que es vinculen a objectes mitjançant noms d'instància continguts en la biblioteca o escenari. Per tal d'entendre millor el concepte, a continuació es mostra la classe utilitzada per als míssils:

```

//Aquest és l'esquelet bàsic d'una classe "as"
package{

//Hem d'importar un alguns "display objects" i events que utilizarem
import flash.display.MovieClip;
import flash.events.*;

//Això bàsicament vol dir que el míssil actuarà com a un clip de pel·lícula
public class misil extends MovieClip{

//Haurem de definir algunes variables al codi de la pel·lícula que veurem al següent punt
//Aquí defineix que l'objecte míssil actuarà com l'arrel del document per tal de poder-hi
fer referència dins de la classe
private var _root:Object;

//Velocitat del míssil
private var speed:int = 10;

//Aquesta funció es ficarà en marxa cada cop que aparegui un míssil al escenari

```

```
public function misil(){

//Afegirem events a la classe
//Funcions que només actuaran quan s'afegeixi un míssil
addEventListener(Event.ADDED, comensarClasse);

//Funció que actuarà en l'"enter frame"
                                //Funció que actuarà en l'"enter frame"
addEventListener(Event.ENTER_FRAME, eFrame);
}
private function comensarClasse (event:Event):void{
    _root = MovieClip(root);
}
private function eFrame(event:Event):void{

//Movem el míssil cap amunt
y += speed

//Movem el míssil cap amunt                                y            -=
speed;//Fent desaparèixer el míssil si surt del marge
if(this.y < -1 * this.height){
removeEventListener(Event.ENTER_FRAME, eFrame);
_root.removeChild(this);
}
                                }
                                _root.bulletContainer.removeChild(this);
                                }
```

El següent element a programar són els míssils que volem que la nau dispari al prémer la tecla *space*. Per a aquesta acció s'han utilitzat les classes "as". Les classes dintre del llenguatge *Actionscript* són codis externs a la pel·lícula que es vinculen a objectes mitjançant noms d'instància continguts en la biblioteca o escenari. Per tal d'entendre millor el concepte, a continuació es mostra la classe utilitzada per als míssils:

```
//Aquest és l'esquelet bàsic d'una classe "as"
package{

//Hem d'importar un alguns "display objects" i events que utilizarem
importflash.display.MovieClip;
import flash.events.*;

//Això bàsicament vol dir que el míssil actuarà com a un clip de pel·lícula
public class misil extends MovieClip{

//Haurem de definir algunes variables al codi de la pel·lícula que veurem al següent punt
//Aquí defineix que l'objecte míssil actuarà com l'arrel del document per tal de poder-hi
fer referència dins de la classe
private var _root:Object;

//Velocitat del míssil
private var speed:int = 10;

//Aquesta funció es ficarà en marxa cada cop que aparegui un míssil al escenari
public function misil(){

//Afegirem events a la classe
//Funcions que només actuaran quan s'afegeixi un míssil
addEventListener(Event.ADDED, comensarClasse);

//Funció que actuarà en l'"enter frame"
```

```
                                //Funció que actuarà en l'"enter frame"
addListener(Event.ENTER_FRAME, eFrame);
}
private function comensarClasse (event:Event):void{
    _root = MovieClip(root);
}
private function eFrame(event:Event):void{

//Movem el míssil cap amunt
y += speed

//Movem el míssil cap amunt                                y            -=
speed;//Fent desaparèixer el míssil si surt del marge
if(this.y < -1 * this.height){
removeEventListener(Event.ENTER_FRAME, eFrame);
_root.removeChild(this);
}
                                }
                                _root.bulletContainer.removeChild(this);
                                }
}
```

Després d'haver creat la classe per als míssils ens falta afegir el codi referent a la classe com algunes variables dins de la pel·lícula:

```
//Comprovar si cTime ha arribat al límit o no
    if(cTime < cLimit){
        cTime ++;
    } else {
        //Si ho fa, permetre disparar
        permetre_disparar = true;
        // reset cTime
        cTime = 0;
    }

//Comprovar si la tecla space està premuda i disparar està permès (Dins de la funció
“comprovar_tecles_amunt”)
    if(event.keyCode == 32 && permetre_disparar){

//No permetre al usuari disparar per un moment
        permetre_disparar = false;

//Declarant variable per crear nou míssil
var newmisil:misil = new misil();

//Definint coordenades del míssil per a que neixi al mig de la nau
newmisil.x = nau.x + nau.width/2 - newmisil.width/2;
newmisil.y = nau.y;

//Llavors afegim el míssil al escenari
addChild(newmisil);
    }
}
```


Un cop escrit això ja es pot disparar míssils pitjant la tecla *space*. Ara cal crear els enemics. En aquest cas es crea un clip de pel·lícula amb un estel de cada color en 6 fotogrames diferents el que dona un efecte d'estel de l'espai al més pur estil videojoc. Per als enemics també crearem una classe "as". Fixant-s'hi es veu que la classe per als enemics és quasi idèntica a la dels míssils però canviant coes puntuals.

```
package{

//Hem d'importar un alguns "display ojects" i events que utilitzarem
import flash.display.MovieClip;
import flash.events.*;

//Això bàsicament vol dir que el míssil actuarà com a un clip de pel·lícula
public class enemic extends MovieClip{

//Haurem de definir algunes variables al codi de la pel·lícula que veurem al següent punt
//Aquí defineix que l'objecte enemic actuarà com l'arrel del document per tal de poder-hi
fer referència dins de la classe
private var _root:Object;

//Velocitat del enemic
private var speed:int = 10;

//Aquesta funció es ficarà en marxa cada cop que aparegui un míssil al escenari
public function misil(){

//Afegirem events a la classe
//Funcions que només actuaran quan s'afegeixi un míssil
addEventListener(Event.ADDED, comensarClasse);

//Funció que actuarà en l'"enter frame"
addEventListener(Event.ENTER_FRAME, eFrame);
}
```

```
private function comensarClasse (event:Event):void{
    _root = MovieClip(root);
}
private function eFrame(event:Event):void{

//Movem el míssil cap amunt
y += speed;

//Fent desaparèixer el míssil si surt del marge
if(this.y < -1 * this.height){
removeEventListener(Event.ENTER_FRAME, eFrame);
_root.removeChild(this);
}
```

Un cop tenim la classe dels enemics feta toca incorporar les variables i el codi necessari per fer aparèixer enemics a la pel·lícula.

```
//Quan de temps abans de crear un altre enemic
var enemyTime:int = 0;

//Quan temps necessari per fer un enemic nou
//Ha de ser més que el rang de dispar o sinó serà impossible matar tots els enemics
var enemyLimit:int = 12;

//Afegint enemics al escenari
if(enemyTime < enemyLimit){

//Si el temps no s'ha assolit, incrementem de més
enemyTime ++;
} else {
```

```
//Definim una variable que contindrà el nou enemic
var newEnemy = new enemy();

//ficant l'enemic fora de l'escenari quan el creem
newEnemy.y = 1 * newEnemy.height;

//Fem un random de les coordenaeds x
//La funció "int" actuarà igual que la "Math.floor" però una mica més ràpid
newEnemy.x = int(Math.random()*(stage.stageWidth - newEnemy.width));

//Llavors afegim l'enemic a l'escenari
addChild(newEnemy);

//reset el enemyTime
enemyTime = 0;
}
```

Ja s'ha aconseguit fer aparèixer estels des de la part de dalt de l'escenari. Ara el següent pas és crear la funció per a que cada míssil col·lisió amb cada estel. Per això es pot fer-ho pensant que cada enemic col·lisionarà amb cada estel o que cada estel col·lisionarà amb cada míssil. En aquest cas es fa de la manera que cada enemic col·lisiona amb cada estel. També crearem una funció per tal de incrementar una puntuació cada cop que es xoca un míssil amb un estel. Per tal de dur a terme aquesta acció, la col·lisió, haurem de crear un clip de pel·lícula que contingui totes els míssils anomenat "bulletContainer". Aquest clip de pel·lícula el definim a la capa accions de l'aplicació.

```
//Aquest clip de pe·lícula contindrà tots els míssils
var bulletContainer:MovieClip = new MovieClip();
addChild(bulletContainer)
```

Ara cal definir el xoc entre els objectes en la classe per als estels de la següent forma:

```
//Comprovant si l'enemic toca algun míssil
//S'ha de ficar en marxa una iteració perquè hi hauran múltiples míssils
for(var i:int = 0;i<_root.bulletContainer.numChildren;i++){
//numChildren és la quantitat de míssils dins del contenidor

//Es defineix una variable per al míssil el qual li farem un test de xoc
var bulletTarget:MovieClip = _root.bulletContainer.getChildAt(i);

//Fem el test de xoc
if(hitTestObject(bulletTarget)){

//Elimina l'enemic si toca el míssil
removeEventListener(Event.ENTER_FRAME, eFrame);
_root.removeChild(this);

//També elimina el míssil i els seus "listeners"
_root.bulletContainer.removeChild(bulletTarget);
bulletTarget.removeListeners();
```

```
//augmentar puntuació
_root.score += 100;
}
}
```

S'ha fet la funció de xoc però cal canviar un parell de coses abans per a que no ens retorni error el programa. També cal aplicar un text dinàmic al escenari vinculat a la puntuació per tal de poder-la visualitzar. Així aplicarem el codi següent que falta.

```
//Per fer desaparèixer els míssils de l'escenari cal afegir a la classe míssil
removeEventListener(Event.ENTER_FRAME, eFrame);
_root.bulletContainer.removeChild(this);

//Per fer aparèixer els míssils al disparar cal afegir a la funció aprèixer
bulletContainer.addChild(newmisil);

//Puntuació del jugador
var score:int = 0;

//Actualitzant el text dinàmic de la puntuació (Dins de la funció "FuncionEframe")
txtScore.text = 'Score: '+score;
}
```

A l'hora de crear el xoc entre objectes en la classe dels enemics, s'ha creat una funció "removeListeners" que s'haurà de definir tan a la classe dels míssils com a la dels enemics.

```
//Aquest funció esborrarà la funció "eFrame"
public function removeListeners():void{
removeEventListener(Event.ENTER_FRAME, eFrame);
}
```

El joc ja està gairebé acabat, ara se li aplica un efecte d'explosió als enemics quan el míssil els toca. Per això s'ha creat un clip de pel·lícula que conté l'animació de l'explosió i una nova classe per a aquest. El codi de la classe té aquest aspecte.

```
package {

import flash.display.MovieClip;
import flash.display.Stage;
import flash.events.Event;

public class explosio extends MovieClip{

//Variable que fa referència a l'escenari
private var stageRef:Stage;

//Funció que servirà per situar la explosió
public function explosio(stageRef:Stage, x:Number, y:Number){
this.stageRef = stageRef;
this.x = x;
this.y = y;
addEventListener(Event.ENTER_FRAME, loop, false, 0, true);}

//Activa la funció d'esborrar l'explosió un cop l'animació que conté arriba a l'últim
fotograma
private function loop(e:Event){
if (currentFrame == totalFrames)
removeSelf();
}

//Funció per a esborrar l'explosió
function removeSelf() : void{
removeEventListener(Event.ENTER_FRAME, loop);
```

```
if (stageRef.contains(this))
stageRef.removeChild(this);
    }

}

}
```

Després de definir la classe de l'explosió, caldrà afegir a la classe dels enemics que és on tenim les funcions de xoc, una línia per afegir l'explosió en el lloc que xoquen el míssil i l'enemic.

```
stage.addChild(new explosio(stage, x, y));
```

En aquest punt, ja s'ha creat un joc per a disparar míssils, matar l'enemic i que aquest exploti. L'últim pas serà fer que s'acabi el joc quan un enemic xoqui amb la nau i fer aparèixer un menú final per a proposar tornar jugar o entrar al *portafoli*, amb un marcador de la puntuació final. Aquest és el codi necessari per a acabar el joc.

```
//Assignant valor "false" a la variable "Game Over" a la capa accions de la pel·lícula
var gameOver:Boolean = false;
//Test de xoc amb la nau
if(hitTestObject(_root.nau)){

//Perdre el joc i enviar al fotograma amb la etiqueta "lose"
_root.gameOver = true;
_root.gotoAndStop('lose');
}
if(_root.gameOver){
removeEventListener(Event.ENTER_FRAME, eFrame);
this.parent.removeChild(this);
}
```

Menú de selecció

Ara el codi que ens falta per a donar per conclouïa la *Intro* és el codi sotmès dins del fotograma del menú. Dins d'aquest menú haurem de crear 4 botons: "Tornar a jugar" i tres altres botons referents a escollir el llenguatge amb el qual vols visualitzar el *portafoli*. També caldrà crear un requadre de text dinàmic per a la puntuació final. I aquest serà el codi:

```
//Parar en aquest fotograma sense avançar
stop();

//Definir les funcions dels dos botons
btTornar.addEventListener(MouseEvent.CLICK, tornar);

//Tornar al fotograma del joc
function tornar(event:MouseEvent):void{
gotoAndStop(321);
stage.removeEventListener(MouseEvent.CLICK, tornar);
}

btEntrar.addEventListener(MouseEvent.CLICK, entrar);
//Entrar a la pròxima escena contenidora del portafoli
function entrar(event:MouseEvent):void{
nextScene(); //S'entra al portafoli amb anglès, el qual està en la pròxima escena
}

//Es crea una variable tipus Loader per carregar una pel·lícula .swf externa
var carregacat:Loader =new Loader() ;
addChild(carregacat) ;

//Es crea un listener que escolta al botó "bt_catala"
bt_catala.addEventListener(MouseEvent.CLICK, entrar_catala);
```



```
//Es defineix la funció per a carregar la pel·lícula externa "pepfoliocat" i les sves
coordenades
function entrar_catala(e:MouseEvent):void{
carregacat.load(new URLRequest("pepfoliocat.swf"));
carregacat.x = 0
carregacat.y = 0
}
//Es crea una variable tipus Loader per carregar una pel·lícula .swf externa
var carregacast:Loader =new Loader() ;
addChild(carregacat) ;

//Es crea un listener que escolta al botó "bt_castella"
bt_catala.addEventListener(MouseEvent.CLICK, entrar_castella);

//Es defineix la funció per a carregar la pel·lícula externa "pepfoliocat" i les sves
coordenades
function entrar_castella(e:MouseEvent):void{
carregacast.load(new URLRequest("pepfoliocat.swf"));
carregacast.x = 0
carregacast.y = 0

}

//Mantenir la nau sense poder-la moure
nau.removeEventListener(Event.ENTER_FRAME, funcionsEFrame);

//Puntuació final
txtFinalScore.text = 'Final Score:'+score;
```

Així s'haurà acabat el procés de programació pel que fa la *Intro* de l'aplicació, seguidament començarem a programar el que es el *portafoli* i tot el seu contingut.

4.3.2. Pre-càrrega

L'escena que apareix sempre després d'entrar al *portafoli* del llenguatge escollit serà la de pre-càrrega. Aquesta escena sorgeix quan la connexió de l'usuari requereix d'un cert temps per carregar l'aplicació. En aquesta s'hi disposen dos simples rectangles de colors: un és el de fons i l'altre va creixent respecte les coordenades de les x ensenyant el procés de càrrega. També es visualitza amb format text el tant per cent del procés de càrrega. El següent codi és l'establert en la capa accions de l'escena:

```
//Aturar en aquest fotograma
stop();

//Es creen dos variables de tipus Number per a medir la carrega i els eu tant per cent
var carregat:Number;
var percent:Number;

//Es crea un listener per a la carrega
barra_carrega.addEventListener( Event.ENTER_FRAME, progres_carrega );

//Es defineix una funció per al procés de carrega
function progres_carrega(e:Event):void{

//Se li dona un valor a la variable "carregat" dels bytes carregats dividit per els bytes totals
carregat = stage.loaderInfo.bytesLoaded / stage.loaderInfo.bytesTotal;

//Se li dona el valor del tant per cent carregat a la variable "percent"
percent = Math.round(carregat * 100);

//S'escala la coordenaada x de la barra de l'escenari amb nom d'instancia "barra_carrega"
barra_carrega.scaleX = carregat;

//S'ordena que al requadre de text dinàmic s'hi visualitzi el tant per cent del procés de
càrrega
```

```
centcarrega_info.text = "Loading... " + percent + "%";

//Si la variable "percent" arriba a un valor de 100 s'esborra el listener i s'avança a la
següent escena
if( percent == 100 ){
barra_carrega.removeEventListener( Event.ENTER_FRAME, progres_carrega );
play();
}
}
```

4.3.1. Menú principal i contingut

Aquesta segurament és la part més complicada de tot el codi *Actionscript* de l'aplicació. Per tal de fer més entenedor l'esquema de la programació del *portafoli* s'explicarà el sistema de com ha estat creada aquesta escena.

Els botons de menú s'han creat a partir d'un de ja dissenyat prèviament que fa de plantilla. Aquests botons han sigut afegits a una variable contenidora i situats a l'escenari a partir d'una iteració i d'un sistema de taules (*array*). Per a col·locar els diferents títols dels botons hem utilitzat el sistema de taules combinat amb el mètode *Split()*. Aquest mètode divideix un objecte de tipus *String* en subcadenaes separant-lo en el punt que apareix el paràmetre delimitador especificat i torna les subcadenaes en un *array*.

La línia del temps es divideix en quatre àrees diferents marcades per etiquetes: "botons", "dins", "fora" i "seccions" per orde d'aparició. L'etiqueta "botons" conté el codi per a la creació i disposició dels botons del menú així com el codi del botó de pantalla completa. També s'hi troba tot el codi referent al re posicionament dels objectes a l'escenari al canviar de mida la finestra i els enllaços externs. L'etiqueta "dins" és la que té el codi amb la transició de quan apareixen els botons a l'escenari cada cop que ens envia cap al menú. L'etiqueta "fora" conté el codi de la transició de quan els botons desapareixen per donar pas al contingut. I l'etiqueta "seccions" conté tot el referent al contingut de les seccions de l'aplicació.

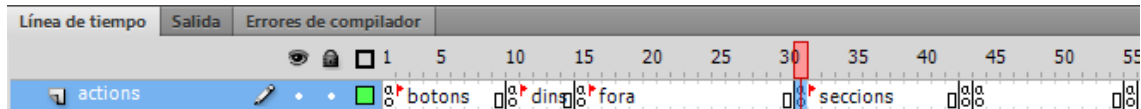


Fig.13 Etiquetes disposades en la línia del temps de l'escena *portafoli*.

Botons

Un cop explicat això es mostra el codi per a totes les accions empreses en el *portafoli*. Es comença amb el codi referent a la creació dels botons del menú i la seva disposició en l'escenari. En l'etiqueta “botons” s'estableix el següent codi:

```
//S'amaga el botó del menú disposat a l'escenari
boto_ppl.visible = false;

//Es crea una taula per a les etiquetes (títols) dels botons del menú
var menu_label:Array = new Array("Profile", "Web&Graphics", "Video&Animation",
"Contact");
//El numero de botons que es crearan usen la propietat "length"
var total:int = menu_label.length;
//Espaiat que utilitzarem al for
var espai:int = 230;
var i:int = 0;
//Variable del control de pagina
var seccio:int;
//Contenedor dels botons de menú
var menu_inici:MovieClip = new MovieClip();
//Agreguem el contenidor a l'escenari
stage.addChild(menu_inici);

//Es defineix una iteració per a crear els botons (Quantitat de botons)
for( i = 0; i < total; i++ ){
```

```
//Es defineix les propietats dels botons del menú
var bt = new boto_menu();
bt.x = 140 + (espai) * i;
bt.y = 300;
bt.alpha = 0;
bt.item_no = i;
bt.flashmo_icon.gotoAndStop( i + 1 );

//Es crea una subcadena per a adherir les etiquetes (títols) dels botons
var subcadena:Array = menu_label[i].split("|");
bt.flashmo_button_label.tm_label.text = subcadena[0];
bt.item_url = subcadena[1];
//S'agrega el botó dins del contenidor
menu_inici.addChild(bt);
}
```

Un cop creats els botons es creen les funcions per tal de donar-los-hi interactivitat:

```
//Es defineixen uns "listeners" als botons del menú per tal donar-los interactivitat amb el
ratolí
bt.addEventListener( Event.ENTER_FRAME, bt_enter );
bt.addEventListener( MouseEvent.ROLL_OVER, bt_over );
bt.addEventListener( MouseEvent.ROLL_OUT, bt_out );
bt.addEventListener( MouseEvent.CLICK, bt_clic );

//Funció que respon un valor booleà "true" al posar el ratolí per sobre del botó
function bt_over(e:MouseEvent):void{
e.target.over = true;
}
```

```
//Funció que respon un valor booleà "false" al deixar de passar el ratolí per sobre del
botó
function bt_out(e:MouseEvent):void{
e.target.over = false;
}

//Funció que respon a l'acció de fer clic amb el ratolí
function bt_clic(e:MouseEvent):void{

//Es crea una variable que defineix els botons del menú com a clip de pel·lícula pare
var mc = e.target.parent;

//Es crida la funció "canvia_secció"
canvia_seccio(mc.item_no);
}

//Funció per activar l'animació del botó quan es passa el ratoí per sobre
function bt_enter(e:Event):void
{

//Variable referent al botó
var mc = e.target;

//Si el ratolí passa per sobre del botó, activem l'animació, sinó reproduïm l'animació
cap enrera
if( mc.over == true )
mc.nextFrame();
else
mc.prevFrame();
}
```

```
//Funció que ens envia a l'etiqueta de la línia del temps "pagina" i ens canvia la pagina  
segons el boto premut  
function canvia_seccio(no:Number):void{  
seccio = no + 1;  
play();  
}
```

Amb aquest codi s'aconsegueix crear els botons, afegir-los a l'escenari i activar les seves funcions. A continuació es mostra i s'explica el codi referent a les transicions dels botons del menú principal amb el mètode *Tweener*. Cal dir que aquest mètode és el que ens ha permès crear dinamisme en quant a interactivitat es refereix.

La classe *Tweener* és una eina molt poderosa que ens permet controlar totes les propietats de les interpolacions de moviment d'un objecte gràfic. Podem definir la propietat que volem animar, la duració de l'animació (en fotogrames o en segons), els valors inicial i final de la propietat i el més interessant, el mètode de suavitzat. El mètode de suavitzat és una funció que s'estableix com a paràmetre i que ens diu quin tipus de moviment s'aplicarà (de rebot, de retard final, etc). A demés, gràcies al detector d'esdeveniments de la classe, podem executar la seqüència en determinats moments.

Tot això ho disposem a l'etiqueta "dins" i "fora" de la línia del temps fent referència a la transició de quan entra el contingut i de la transició de quan es retira el aquest. Per a dur a terme aquesta acció també farem servei d'un nou mètode anomenat *Timer*. Aquesta classe permet crear objectes que s'executen en intervals de temps específics. Per a utilitzar-los només cal crear l'objecte, indicar el numero de repeticions que volem y el interval que els separa. Tot seguit veurem l'exemple que hem fet servir per a les nostres transicions en l'etiqueta "dins".

```
//Aturar en aquest fotograma

stop();

//S'importa els paquets caurina necessaris en les transicions

import caurina.transitions.*;

//Es crea una variable Timer per a crear un interval de temps entre l'aparició de cada
botó

var timer:Timer = new Timer(140);

timer.addEventListener(TimerEvent.TIMER, in_timer);

timer.start();

//Es defineix la funció Timer

function in_timer(te:TimerEvent):void {

//Es crea una variable que conté cada botó que anirà sortint

var mc = MovieClip( menu_inici.getChildAt(timer.currentCount - 1) );

//S'executa la transició Tween tipus "easeOutExpo" per als botons

Tweeners.addTween( mc, {alpha: 1, y: 150, time: 0.8, transition:"easeOutExpo"} );

//Si l'interval de temps és igual o major al nombre de botons que han apregut, eliminem
el Timer

if( timer.currentCount >= menu_inici.numChildren )

timer.removeEventListener(TimerEvent.TIMER, in_timer);

}
```


Es pot veure que al executar-se la funció *Timer*, cada 140 milisegons, s’executa una transició del tipus “*easeOutExpo*” a l’objecte “*mc*” mitjançant la classe *Tweener*. Les propietats de la transició estan definides dins dels parèntesis; així l’objecte és disposarà a l’escenari passant a tenir valor *alpha* (transparència) =1, en el punt 150 de la coordenada y, en un interval de 0.8 segons.

Cal dir que per a fer possible aquest tipus d’animacions és imprescindible descarregar el paquet de transicions i disposar-los dins de la carpeta de l’aplicació. També cal llavors, importar-los dins del fotograma on s’executaran. A continuació es mostra els tipus de transicions disponibles:

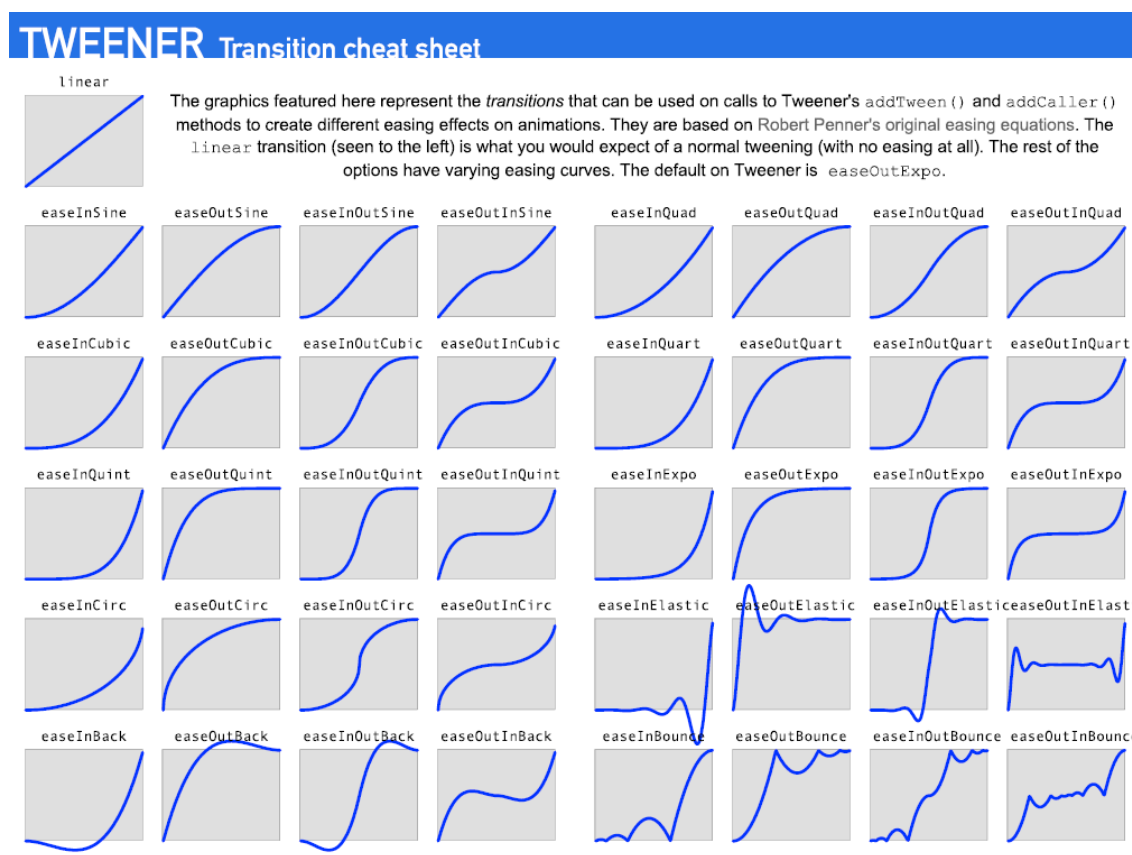


Fig. 14. *Full de transicions *Tweener*. Els gràfics mostrats representen les transicions que poden ser usades mitjançant a la crida dels mètodes “*Tweeners*” *addTween()* i *addCaller()* per a crear diferents efectes de moviment a les animacions. Estan basades en les equacions de moviment originals de *Robert Penner*. La transició lineal (a dalt a l’esquerra) és el que t’esperaries d’una interpolació normal. Totes les altres content moviment curvilini. La transició per defecte en la classe *Tweener* és la “*easeOutExpo*”.

Un cop vist el codi per a la transició d'entrada i entès com funciona, es mostra el codi utilitzat per a la transició de sortida. Es pot comprovar que és molt semblant al d'entrada.

```
//Aturar en aquest fotograma
stop();

//Es crea el Timer
timer = new Timer(80);
timer.addEventListener(TimerEvent.TIMER, out_timer);
timer.start();

//Es defineix la funció Timer
function out_timer(te:TimerEvent):void {
//Es crea una variable que conté cada botó que anirà sortint
var mc = MovieClip( menu_inici.getChildAt(timer.currentCount - 1) );

//S'executa la transició Tween tipus "easeInStrong" per als botons
Tweeners.addTween( mc, {alpha: 0, y: 800, time: 1, transition:"easeOutExpo" } );

//Si l'interval de temps és igual o major al nombre de botons que han apregut, eliminem el
Timer i seguim en la línia del temps
if( timer.currentCount >= menu_inici.numChildren ){
play();
timer.removeEventListener(TimerEvent.TIMER, out_timer);
}
}
```

Contingut

Al haver creat els botons del menú principal i les seves transicions és hora de veure com funciona el codi que conté el clip de pel·lícula on hi ha cada secció. El sistema per a dirigir-se al contingut està fet de manera que quan es clica al botó del menú principal, l'usuari es traslladat a l'etiqueta “seccions” on entrarem mitjançant una transició a la

secció determinada per a cada botó del menú. Les transicions d'entrada i sortida a les pàgines del contingut estan creades manualment com a interpolacions de moviment en la línia del temps. Funciona de manera que quan s'entra al contingut, s'envia l'usuari a dins de l'etiqueta "seccions" i l'atura en aquesta. Aquesta etiqueta en la línia del temps conté un botó per a retornar al menú principal al pitjar-lo. Un cop realitzada aquesta acció la línia del temps es fica en marxa reproduint la interpolació de moviment referent a la transició de sortida de secció, fins arribar a l'últim fotograma num.55 (veure fig.9) de l'escena que ens reenvia a l'etiqueta "dins" per a donar entrada als botons del menú principal de nou i esborra el botó de tornar al menú principal de l'escenari.

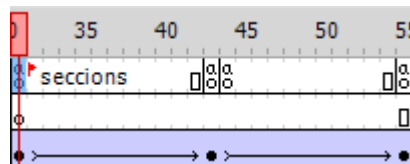


Fig. 15. Etiqueta "seccions" i interpolació de moviment referent a les pàgines de contingut.

Gairebé tot el codi establert en el fotograma d'accions de l'etiqueta "seccions" fa referència al botó que es crea dins de les pàgines de contingut amb la funció de tornar al menú principal. Aquest botó també es crea mitjançant la plantilla amb la que s'han creat tots els del menú principal. Com es veu a continuació, el codi és similar.

```
//Es defineix l'etiqueta (títol) del botó a mostrar com a "Return to Main Menu"
bt_tornar.flashmo_button_label.tm_label.text = "Return to Main Menu";

//Es creen els listeners que escoltaran al botó "bt_tornar"
bt_tornar.addEventListener( Event.ENTER_FRAME, tornar_enter );
bt_tornar.addEventListener( MouseEvent.ROLL_OVER, tornar_over );
bt_tornar.addEventListener( MouseEvent.ROLL_OUT, tornar_out );
bt_tornar.addEventListener( MouseEvent.CLICK, tornar_to_menu );

//S'afegeix el botó "bt_tornar" a l'escenari
addChild(bt_tornar);

//Se li aplica al botó "bt_tornar" una transició Tween tipus "easeOutExpo" al aparèixer
Tweener.addTween( bt_tornar, { alpha: 1, y: 130, time: 0.8, transition:"easeOutExpo" } );

//S'avança en la línia del temps al fer clic al botó "bt_tornar"
function tornar_to_menu( e:MouseEvent ):void{
play();
}

//Funció que respon un valor booleà "true" al posar el ratolí per sobre del botó
function tornar_over(e:MouseEvent):void{
e.target.over = true;
}

//Funció que respon un valor booleà "false" al deixar de passar el ratolí per sobre del botó
function tornar_out(e:MouseEvent):void{
e.target.over = false;
}

//Funció per activar l'animació del botó quan es passa el ratolí per sobre del botó
```

```
"bt_tornar"  
function tornar_enter(e:Event):void{  
//Variable referent al botó "bt_tornar"  
var mc = e.target;  
  
//Si el ratolí passa per sobre del botó, activem l'animació, sinó reproduim l'animació  
cap enrera  
if( mc.over == true )  
mc.nextFrame();  
else  
mc.prevFrame();  
}
```

Ara fa falta veure el codi escrit en l'últim fotograma (num.55) que retorna a l'etiqueta "dins" i esborra el botó "tornar".

```
//Al arribar en aquest fotograma anar a l'etiqueta "in" de la línia del temps  
gotoAndPlay("dins");  
  
//Eliminar de l'escenari el botó "bt_tornar"  
removeChild(bt_tornar);
```

Amb aquest últim codi s'ha pogut veure l'esquema de programació del que és el menú principal i la seva vinculació amb el contingut de les diferents seccions.

Ara toca veure el codi utilitzat en les diferents pàgines de continguts, és a dir, en l'interior de les diferents seccions de l'aplicació. Aquestes pàgines estan disposades totes dins d'un clip de pel·lícula amb nom d'instància "contingut". Cadascuna ocupa un fotograma amb el contingut repartit en varies capes (veure fig.10).

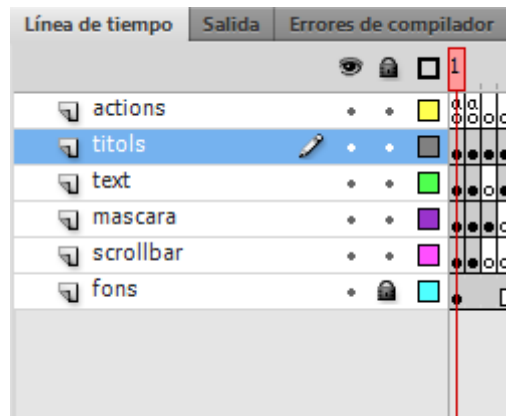


Fig.16. Interior del clip de pel·lícula “contingut” amb les diferents pàgines de contingut ocupant 4 fotogrames com seccions hi ha.

En la UC5 “Perfil” el codi establert fa referència a un *link* que ens envia a la visualització del CV del candidat en format *.pdf* i amb possibilitat de descarrega. El codi s’estableix dins del clip de pel·lícula amb nom d’instància “*ct_page1*” i és el següent:

```
//Enllaç extern a .pdf
bt_downloadcv.addEventListener(MouseEvent.CLICK, mostrarcv);

function mostrarcv(e:MouseEvent):void{
    navigateToURL(new URLRequest("CV_JosepBorrasYEnglish.pdf"));
}
```

En la UC6 “Web&Gràfics” disposem d’una funció per fer créixer les imatges al fer clic sobre elles. El codi necessari per dur a terme aquesta funció es troba a dins del clip de pel·lícula amb nom d’instància “*ct_page2*”. Cal destacar la utilització del mètode interpolació de moviment (*Tween*), però en aquest cas en forma de codi i no manualment.

```
//Importar la classe d'animació Tween
import fl.transitions.Tween;
import fl.transitions.easing.*;

//Objectes Tween per animar la transició
var tweenX:Tween;
var tweenY:Tween;

//Reduir mida
var size:Number = 1;

//Velocitat de l'animació fer créixer/reduir
var speed:Number = 0.5;

// Filtre per donar una sombra caiguda
var dsf:DropShadowFilter = new DropShadowFilter(4,45,0,1,4,4);
// Aplicar el filtre

imatgeweb.filters = [dsf];
imatgeweb1.filters = [dsf];

//S'escala a mida reduïda
imatgeweb.scaleX = size;
imatgeweb.scaleY = size;
imatgeweb1.scaleX = size;
imatgeweb1.scaleY = size;
imatgeweb2.scaleX = size;
imatgeweb2.scaleY = size;
logo_delve.scaleX = size;
logo_delve.scaleY = size;

//Es crea per a cada imatge un listener amb funció quan es fa clic
imatgeweb.addEventListener(MouseEvent.CLICK, growHandler);
```

```
imatgeweb1.addEventListener(MouseEvent.CLICK, growHandler);
imatgeweb2.addEventListener(MouseEvent.CLICK, growHandler);
logo_delve.addEventListener(MouseEvent.CLICK, growHandler);

//S'estableix el mode botó als clips de pel·lícula
imatgeweb.buttonMode = true;
imatgeweb1.buttonMode = true;
imatgeweb2.buttonMode = true;
logo_delve.buttonMode = true;

// Funció pel primer cop que es fa clic (des d'imatge petita)
function growHandler(event:MouseEvent):void {

var mc:DisplayObject = event.currentTarget as DisplayObject;

imatgeweb.removeEventListener(MouseEvent.CLICK, growHandler);
imatgeweb.addEventListener(MouseEvent.CLICK, shrinkHandler);
imatgeweb1.removeEventListener(MouseEvent.CLICK, growHandler);
imatgeweb1.addEventListener(MouseEvent.CLICK, shrinkHandler);
imatgeweb2.removeEventListener(MouseEvent.CLICK, growHandler);
imatgeweb2.addEventListener(MouseEvent.CLICK, shrinkHandler);
logo_delve.removeEventListener(MouseEvent.CLICK, growHandler);
logo_delve.addEventListener(MouseEvent.CLICK, shrinkHandler);

//Cridar la funció "creixer" a l'objecte clicat
    growIt(mc);
}

// Funció pel segon que es clica (Des d'imatge gran)
function shrinkHandler(event:MouseEvent):void {
    var mc:DisplayObject = event.currentTarget as DisplayObject;

imatgeweb.addEventListener(MouseEvent.CLICK, growHandler);
```



```
imatgeweb.removeEventListener(MouseEvent.CLICK, shrinkHandler);
imatgeweb1.removeEventListener(MouseEvent.CLICK, shrinkHandler);
imatgeweb1.addEventListener(MouseEvent.CLICK, growHandler);
imatgeweb2.removeEventListener(MouseEvent.CLICK, shrinkHandler);
imatgeweb2.addEventListener(MouseEvent.CLICK, growHandler);
logo_delve.removeEventListener(MouseEvent.CLICK, shrinkHandler);
logo_delve.addEventListener(MouseEvent.CLICK, growHandler);

//Cridar la funció per reduir quan es clica l'objecte
    shrinkIt(mc);
}
//Fer créixer objecte
function growIt(mc:DisplayObject):void {
    tweenX = new Tween(mc,"scaleX",null,mc.scaleX,3,speed,true);
    tweenY = new Tween(mc,"scaleY",null,mc.scaleY,3,speed,true);
}

// Reduir objecte
function shrinkIt(mc:DisplayObject):void {
    tweenX = new Tween(mc,"scaleX",null,mc.scaleX,size,speed,true);
    tweenY = new Tween(mc,"scaleY",null,mc.scaleY,size,speed,true);
}
```

En la UC7 “Vídeo&Animació” s’hi estableix un reproductor de vídeo mitjançant una plantilla que ens ofereix Lisa Larson-Kelley a la pàgina oficial d’*Adobe*. La feina de producció aquí és la de convertir tots els vídeos a format *.flv* per tal d’afegir-los a la plantilla.

A la UC8 “Contacte” s’hi disposa un clip de pel·lícula que conté el formulari amb els requadres de text dinàmic que l’usuari ha d’omplir. En aquest clip de pel·lícula s’hi disposa també el codi necessari per a comunicar-se amb el candidat via correu electrònic. El codi per a formularis de contacte en llenguatge *actionscript 3.0* no varia molt d’un model o altre i aquest és el que s’ha fet servir:

```
//Es deixen en blanc els requadres de text dinàmic
nom_contacte.text = email_contacte.text = tema_contacte.text =
miss_contacte.text = status_miss.text = "";

//Es crea listener per als botons enviar i esborrar
boto_enviar.addEventListener(MouseEvent.CLICK, enviar);
boto_reset.addEventListener(MouseEvent.CLICK, reset);

//Es crea una variable Timer
var timer:Timer;

//Es creen les variables per a la sol·licitud d'enviar email mitjançant PHP
var var_load:URLLoader = new URLLoader;
var URL_request:URLRequest = new URLRequest( "send_email.php" );
URL_request.method = URLRequestMethod.POST;

//Funció del botó "enviar"
function enviar(e:MouseEvent):void{

//Si els requadres de text dinàmic estan buits
if( nom_contacte.text == "" || email_contacte.text == "" ||tema_contacte.text == "" ||
miss_contacte.text == "" ){

//Mstrar aquest text al requadre "status miss"
status_miss.text = "Please fill up all text fields.";
}

//Sinó si l'email dona't és invàlid
else if( !validate_email(email_contacte.text) ){

//Mostrar aquest text al requadre "status miss"
status_miss.text = "Please enter the valid email address.";
}
else{
```

```
//Sinó ensenyar aquest
status_miss.text = "sending...";

//Es crea variable del text que es mostrarà sobre la informació de l'usuari
var email_data:String = "name=" + nom_contacte.text
    + "&email=" + email_contacte.text
    + "&subject=" + tema_contacte.text
    + "&message=" + miss_contacte.text;

var URL_vars:URLVariables = new URLVariables(email_data);
URL_vars.dataFormat = URLLoaderDataFormat.TEXT;

URL_request.data = URL_vars;
var_load.load( URL_request );
var_load.addEventListener(Event.COMPLETE, receive_response );
}
}

//Funció esborrar el text omplert
function reset(e:MouseEvent):void{
nom_contacte.text = email_contacte.text = tema_contacte.text =
miss_contacte.text = status_miss.text = "";
}

//Funció validar l'email de l'usuari
function validate_email(s:String):Boolean
{
var p:RegExp = /(\\w|[_\\.\\-])+@((\\w|-)+\\.)+\\w{2,4}+\\/;
var r:Object = p.exec(s);
if( r == null ) {
return false;}
return true;
}
```

```
}

//Funció per rebre resposta si s'envia el missatge
function receive_response(e:Event):void{
var loader:URLLoader = URLLoader(e.target);
var email_status = new URLVariables(loader.data).success;

if( email_status == "yes" ){
status_miss.text = "Success! Your message was sent.";
timer = new Timer(500);
timer.addEventListener(TimerEvent.TIMER, on_timer);
timer.start();
}
else{
status_miss.text = "Failed! Your message cannot sent.";
}
}

function on_timer(te:TimerEvent):void {
if( timer.currentCount >= 10 {
nom_contacte.text = email_contacte.text = tema_contacte.text =
miss_contacte.text = status_miss.text = "";
timer.removeEventListener(TimerEvent.TIMER, on_timer);
}
}
```

Cal remarcar que tant la UC5 com la UC6 disposen d'un *scrollbar* establert en un clip de pel·lícula. Aquest *scrollbar* el disposem en les UC afegint el següent codi a la seva capa accions:

```
//S'afegeix l'scrollbar a la UC5  
flashmo_sb1.scrolling("ct_page1", "ct_mask1", 0.20);  
  
//S'afegeix l'scrollbar a la UC6  
flashmo_sb1.scrolling("ct_page2", "ct_mask2", 0.20);
```

El codi referent al *scrollbar* l'hem disposat a l'apartat d'annexes ja que és una plantilla agafada d'internet [].

4.3.1. Posicionament i mode *fullscreen*

Posicionament

Un cop exposat el codi de les diferents pàgines de contingut, el codi establert en l'etiqueta “botons” referents al re posicionament dels objectes de l'escenari. Aquest codi serveix per a que l'aplicació es vegi igual i de la millor manera a tots els monitors amb qualsevol resolució. A continuació es mostra el codi:

```
// S'inicialitza la funció
resizeObjects (null);

//Es defineix una funció per a alinear tots els objectes de l'escenari
function resizeObjects (event:Event):void {

//Es creen dos variables definin la mida de la pel·lícula
var sw:Number = stage.stageWidth;
var sh:Number = stage.stageHeight;

//Es posciona cada objecte de l'aplicació
peu.width = sw;
peu.y = sh - peu.height;
mp3_player.y = sh - (mp3_player.height + peu.height + 52);
peu2.x = sw - peu2.width;
peu2.y = sh - (peu2.height + peu.height + 3)
logo.x = (logo.stage.stageWidth / 2) - (logo.width / 2);
logo.y = 70;
fons.width = sw;
fons.height = sh;
bt_bl.x = sw - 118;
bt_bl.y = 10;
bt_fb.x = sw - 48;
bt_fb.y = 10;
bt_ms.x = sw - 82;
bt_ms.y = 10;
bt_pcompleta.x = sw - (bt_pcompleta.width + 10);
bt_pcompleta.y = sh - 90;
}
```

Mode *fullscreen*

Un dels altres codis que es troben a l'etiqueta botons de la línia del temps és el del botó per passar a mode pantalla completa. Un codi molt útil i que ens donarà un servei per a una propietat fonamental per a una aplicació d'aquestes característiques. S'ha aplicat de la següent manera:

```
import flash.display.StageDisplayState;

//Ficar instancia de tipus clip de pel·lícula amb mode botó
bt_pcompleta.buttonMode = true;

//Es defineix una funció per a executar el mode pantalla completa
function goFullScreen():void{
if (stage.displayState == StageDisplayState.NORMAL) {
stage.displayState=StageDisplayState.FULL_SCREEN;
} else {
stage.displayState=StageDisplayState.NORMAL;
}
}

//Es creen els listeners que escoltaran el botó per a passar a mode pantalla completa
bt_pcompleta.addEventListener(MouseEvent.CLICK, pcompleta);
bt_pcompleta.addEventListener(MouseEvent.MOUSE_OVER, mourebt);
bt_pcompleta.addEventListener(MouseEvent.MOUSE_OUT, pararbt);
bt_pcompleta.addEventListener( Event.ENTER_FRAME, enterbt );

//S'activa el mode pantalla completa al fer clic sobre el botó
function pcompleta(event:MouseEvent):void{
goFullScreen();
}
}
```

```
//Funció que respon un valor booleà "true" al posar el ratolí sobre del botó
function mourebt(event:MouseEvent):void{
bt_pcompleta.over = true;
}

//Funció que respon un valor booleà "false" al deixar de passar el ratolí per sobre del botó
function pararbt(event:MouseEvent):void{
bt_pcompleta.over = false;
}

//Funció per activar l'animació del botó quan es passa el ratoí per sobre
function enterbt(e:Event):void{

//Si el ratolí passa per sobre del botó, activem l'animació, sinó reproduim l'animació cap
enrera
if( bt_pcompleta.over == true )
bt_pcompleta.nextFrame();
else
bt_pcompleta.prevFrame();
}
```


4.3.1. Sons

A alguns elements de l'aplicació que tenen la funció de botó, se'ls hi aplica sons en quan arrosseguem punter per sobre seu, o quan hi fem clic. Consisteix en importar els sons que vulguem a la biblioteca i dona'ls-hi un nom de classe en la finestra de les seves propietats.

El codi necessari per a tal acció és el mateix per a tots i té aquesta forma:

```
//Aquest so està situat dins de la funció que detecta quan el punter es situa sobre de  
l'element botó  
function bt_over(e:MouseEvent):void{  
//Es diu al so que es reproduceixi  
var so:Sound = new bubble();  
so.play();
```


5. Postproducció.

En aquest capítol es fa un seguiment del tractament de l'aplicació un cop finalitzat el procés de producció

5.1. Publicació del *portafoli*.

Un cop creada l'aplicació, és moment per a publicar-la a internet per donar-se a conèixer. L'eina utilitzada per a la publicació serà el programa *Adobe Dreamweaver CS5*.

Aquest procés comença a l'hora de publicar l'aplicació des de *Flash*. Publicarem l'arxiu *.fla* a format *.swf* i format *html*. A l'hora de publicar es dona la possibilitat de canviar paràmetres entrant a la "configuració de publicació", en la nostra aplicació ajustarem aquests com es mostra a la figura X.

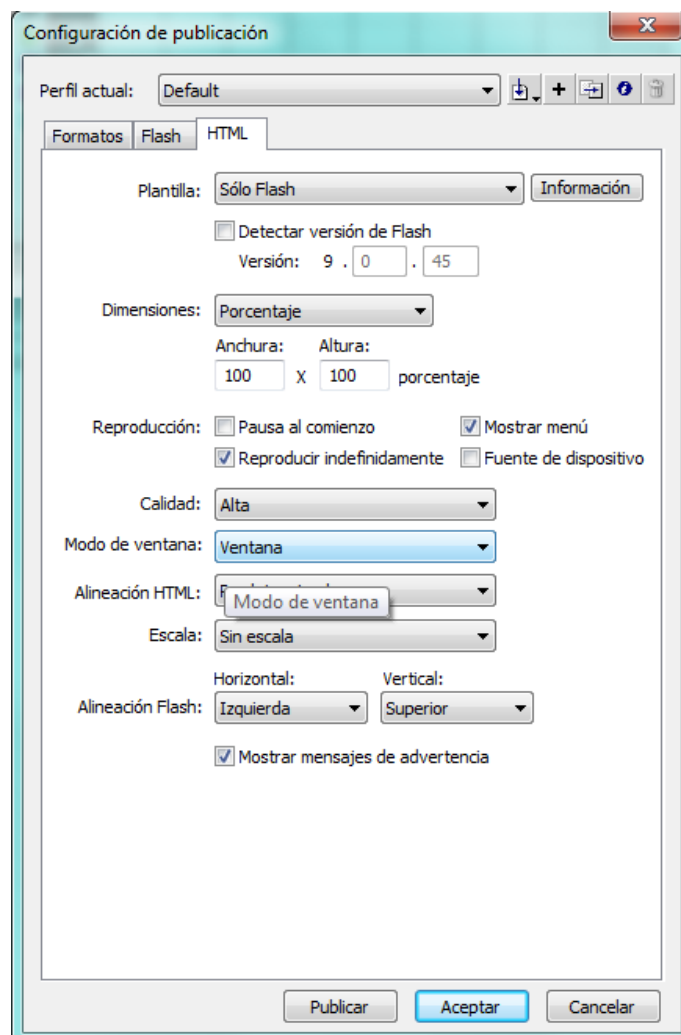


Fig.17. Configuració del codi *html* a publicar.

Un cop es crea el codi *html* de l'aplicació, cal fer una última modificació per tal que ens permeti executar la funció de pantalla completa quan estigui penjada a internet. Simplement cal afegir un paràmetre en el codi el qual serà definit de la següent manera:

```
<param name="allowFullScreen" value="true" />
```

Havent fet aquest ajustament ja es pot passar al *Dreamweaver* per a penjar-ho a internet. El primer que farem un cop dins del programa és crear un nou espai i definir-hi el servidor en el qual es penja. Per a aquesta aplicació s'utilitza el servidor de la EUPMT.

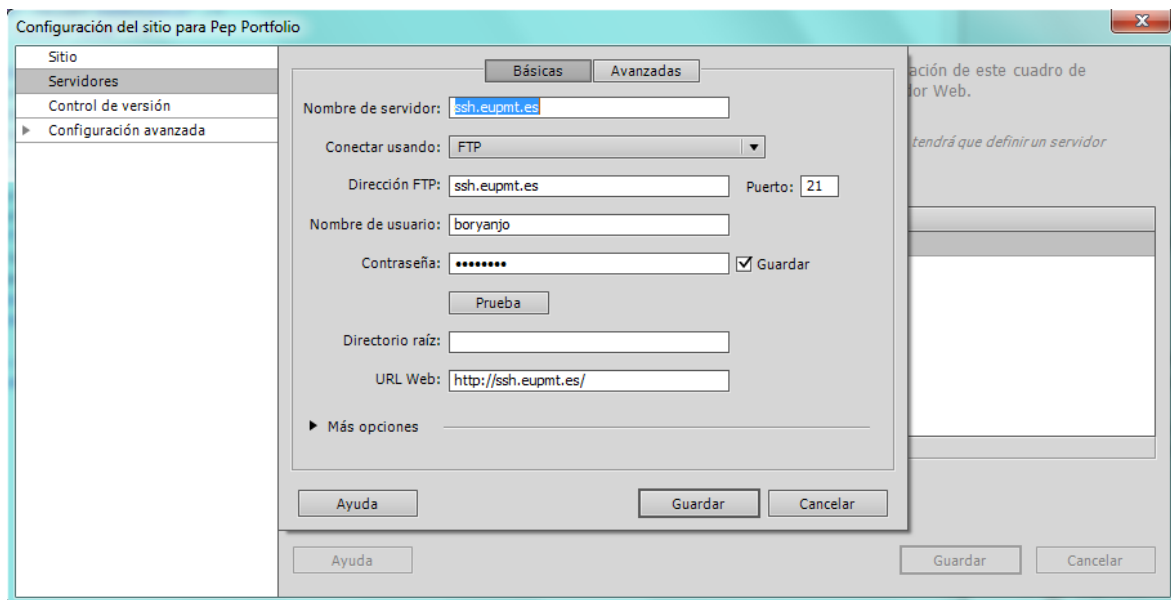


Fig. 18. Configuració de l'espai creat on penjar l'aplicació.

Creat l'espai i la configuració del servidor on es penja, només cal assegurar-se de tenir tots els elements de l'aplicació en una mateix directori arrel i anomenar el document *html* com a "index". El següent pas és prémer el botó amb l'acció de col·locar tot al servidor i ja tindrem l'aplicació penjada a internet (veure fig.X)

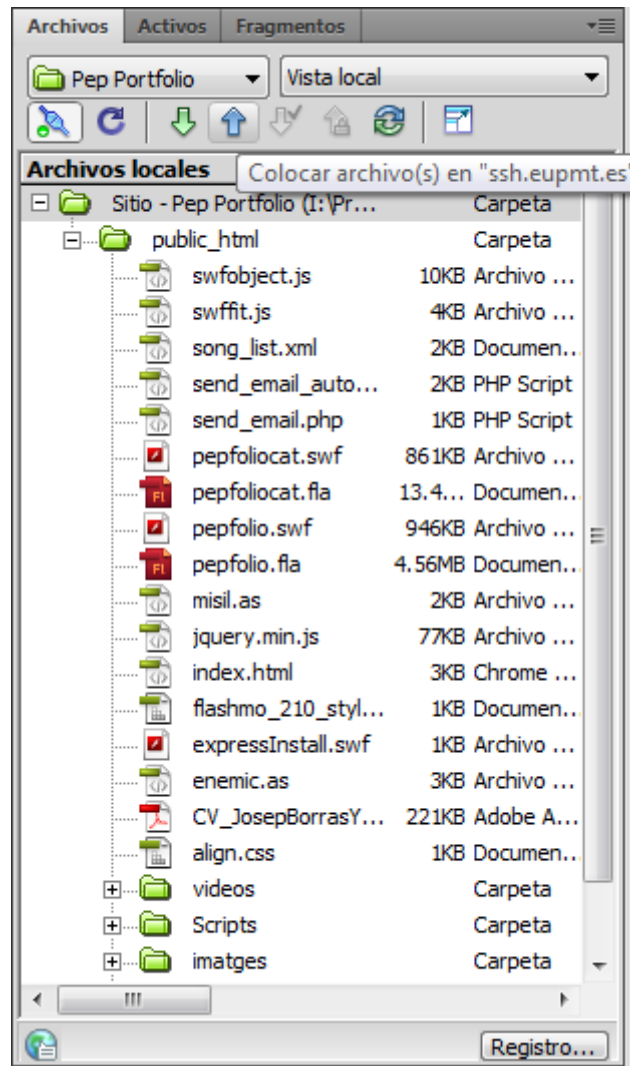


Fig. 19. Colocació de l'aplicació mitjançant *Dreamweaver*.

5.2. Proves d'usabilitat.

Un cop s'ha publicat l'aplicació a Internet, s'ha de fer una revisió de compatibilitat amb navegadors i alguns aspectes relacionats amb la usabilitat. Mitjançant terceres persones, s'ha de deixar que utilitzin l'aplicació per a que posteriorment donin la seva opinió i poder, així, identificar possibles errors i verificar la funcionalitat de tots els enllaços. Això ajuda a millorar els següents aspectes del producte:

- Disseny de l'aplicació i errors de maquetació.
- Possibles enllaços trencats.

- Comprovar que la navegació és intuïtiva.
- Verificar que totes les pàgines contenen títol.
- Mantenir enllaços cap a la pàgina principal en tot moment.
- Verificar que no existeixen imatges amb mala qualitat o mal col·locades.

5.2. Problemes i solucions.

Un dels processos que ha donat més mals de cap a l'hora de crear l'aplicació ha estat el de programació amb *Actionscript 3.0*. És una eina complexa que sense coneixements necessaris pot resultar molt dura de dominar, ja que al mínim error l'aplicació no funciona. La millor solució que s'ha trobat ha estat el manual d'ajuda d'*Adobe* [] i portals web dedicats a aquest llenguatge [[]].

6. Impacte Mediambiental

Aquest projecte no ha necessitat grans quantitats d'energia o material per a la seva realització. Al contrari, el que aconseguim és l'estalvi de recursos. El fet de disposar d'una carta de presentació digital significa l'estalvi de molts currículums i treballs en paper a enviar. El mètode digital es pot considerar com a un molt bon recurs per tal de disminuir la utilització de paper. Tenint en compte la situació actual en el món que vivim, on el nostre pulmó situat a l'Amazones viu moments crítics, es creu que és de gran consideració poder participar en la conservació del medi ambient.

7. Propietat Intel·lectual.

En aquest projecte s'ha fet ús d'un recurs creat per una tercera persona i que ens ha ajudat a millorar la presentació del producte.

Es tracta dels sons, que han estat descarregats del link: www.freesound.org

Aquests sons estan disponibles a internet i són de lliure ús per a qualsevol usuari.

8. Conclusions i ampliacions

8.1. Conclusions

Aquest projecte final de carrera ha assolit els seus objectius un cop finalitzat. És pot dir que s'ha aconseguit crear una *portafoli* digital multimèdia amb un disseny final atractiu i funcional. El producte final conté totes les característiques i propietats requerides per tal de donar pas al món laboral i tenir la possibilitat de presentar en un mercat.

El procés d'elaboració d'un projecte final de carrera d'aquest tipus ha requerit molt d'esforç i sobretot molta paciència. Al tractar-se d'un procés llarg és molt important tenir un bon pla de treball per tal d'evitar al màxim els contratemps. Uns contratemps que inevitablement sorgeixen i s'han de solucionar amb força dedicació. Les ganes i la il·lusió de dur a terme aquest projecte han sigut claus per afrontar-ho tot de la millor manera. Val a dir que una de les millors sensacions que s'han viscut en tot el procés d'elaboració, ha sigut la de l'èxit d'en sortir-se i tirar endavant cada contratemps amb ganes i voluntat.

En quant a l'aspecte tècnic del projecte, es pot valorar la gran utilitat dels programes de la firma *Adobe* per emprendre processos d'aquestes característiques. Uns programes que així com el temps passa ens van ampliant les possibilitats per a crear.

La metodologia de treball empleada, ha estat seqüencial mentre s'anaven avançant els punts anotats en la preproducció, fins a l'hora de programar l'aplicació que ha adoptat una forma més circular o d'espiral, en que ens apareixien problemes i processos a completar.

En conclusió es pot dir que l'elaboració d'aquest projecte final de carrera produït simulant un producte professional, ha permès veure realment el que costa produir des de zero una aplicació d'aquest tipus. Una experiència valorada molt positivament, igual que els resultats obtinguts. Un primer projecte de molts per venir per tal d'assolir l'objectiu més important: seguir aprenent i coneixent conceptes nous dins del món audiovisual/multimèdia.

8.2. Ampliacions

Aquest projecte, disposant de més temps s'anirà ampliant en quan a contingut i disseny a mesura que passi el temps. Es millorarà i s'ampliarà el joc interactiu per tal de donar-li un atractiu més gran. També s'actualitzarà el *portafoli* cada cert temps a mesura que es vagin aprenent noves tècniques producció.

És d'especial importància tenir present que aquest projecte ha servit d'introducció a la creació d'aplicacions multimèdia. Volent dir que també hi haurà una ampliació de treballs a realitzar per part del candidat.

9. Referències.

- [1] Lina Juan i Montse Rabassa. Disseny i Producció d'interactius, Teoria. Llibre de l'assignatura "Disseny i Producció d'interactius"
- [2] help.adobe.com/es_ES/Flash/
- [3] www.adobe.com/devnet/flash/articles/video_playlist.html, Web video template
- [4] abduzeedo.com
- [5] <http://www.flashgametuts.com>, Free flash game tutorials.
- [6] http://www.adobe.com/devnet/actionscript/articles/oop_as3.html, Object-Oriented programming with Actionscript 3.0.
- [7] <http://www.gotoandlearn.com/>, Flash tutorials
- [8] <http://asgamer.com/>, Everything how to make flash games
- [9] <http://www.kirupa.com/>
- [10] <http://www.sargentoweb.com/as3/>
- [11] <http://www.flashmo.com/>
- [12] <http://www.webdesignmo.com/blog/>

Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Graduat en Mitjans Audiovisuals

PORTAFOLI DIGITAL MULTIMÈDIA

Estudi econòmic

JOSEP BORRÀS YÁÑEZ
PONENT: LINA JUAN

PRIMAVERA 2011



TecnoCampus
Mataró-Maresme

Índex.

1. Cost de l'aplicació.....	1
1.1. Cost del material.....	1
1.2. Costos de recursos humans.....	2
1.3. Amortització d'equips i software.....	2
1.4. Despeses indirectes.....	3
1.5. Cost de producció de l'aplicació.....	3
2. Preu de venda en el mercat.....	III

1. Cost de l'aplicació.

En aquest document es mostra un estudi econòmic que simula el que serien les despeses econòmiques relacionades amb l'aplicació creada. L'estudi del pressupost ha considerat els següents factors:

- Les despeses de material.
- Cost de disseny del projecte (hores/dissenyador gràfic).
- Cost administratiu/va (redacció projecte).
- Cost de realització (hores/dissenyador gràfic, hores/programador).
- Despeses d'edifici (llum, telèfon, aigua).
- Cost d'amortització de les eines que s'han de renovar (ordinadors, tabletas gràfiques, material d'oficina, impressores...)
- Altres despeses que s'hagin fet imputables al projecte/treball.

1.1. Cost del material.

Cost del material per al muntatge d'una aplicació multimèdia:

<u>Descripció</u>	<u>Quantitat</u>	<u>Preu unitari (€)</u>	<u>Total (€)</u>
<u>Material informàtic</u>			
Laptop	1	1.500	1.500
Tableta gràfica	1	300	300
Impressora	1	400	400
<u>Material d'oficina</u>			
Paquets de paper	10	5	5
Estilogràfica	5	5	5
TOTAL COST DEL MATERIAL			2.210

1.2. Costos de recursos humans.

<u>Concepte</u>	<u>Hores</u>	<u>Preu/hora (€)</u>	<u>Total (€)</u>
Estudi i documentació (Tècnic audiovisual)	10	20	200
Disseny (Dissenyador gràfic)	30	25	750
Programació (Programador <i>actionsript, html, javascript, css</i>)	40	25	1.000
Redacció memòria (Administratiu)	16	15	240
TOTAL RECURSOS HUMANS			2.190

1.3. Amortització d'equips i software.

<u>Equip utilitzat</u>	<u>Hores d'utilització</u>	<u>Preu/hora</u>	<u>Total</u>
<u>Equips i programari informàtic</u>			
Ordinador	100	0,50	200
Adobe Flash CS5	40	1,00	40
Adobe Photoshop CS5	15	1,00	15
Adobe Illustrator CS5	15	1,00	15
Adobe Dreamweaver CS5	9	1,00	9
Adobe Encore CS5	1	1,00	1
Office	16	1,00	16
TOTAL AMORTITZACIONS			296€

1.4. Despeses indirectes.

El nostre equip humà treballa en unes oficines llogades. Per tant tenim una despesa del cost del lloguer més les despeses d'electricitat, aigua, calefacció etc. En aquestes oficines la despesa més gran és la d'electricitat pel fet que treballem amb ordenadors engegats 20h al dia. En el cost total de la producció del nostre producte incloïem el 15% d'aquestes despeses. Aquest cost s'eleva a 300€ mensuals.

1.5. Cost de producció de l'aplicació.

Costos de material	2.210 €
Costos de recursos humans	2.190 €
Costos d'amortització	296 €
Subtotal	4.696 €
Despeses indirectes (19%)	300€
TOTAL	4996€

2. Preu de venda en el mercat.

El cost total obtingut per la producció de l'aplicació digital multimèdia és d'11.126,5€. Queda clar que en una producció d'aquest tipus hi ha el cost del material que és una gran part del total i que s'amortitza per un gran temps.

Cost desenvolupament	4.996€
Resta del cost material	2.210 €
Cost TOTAL	2.786 €
PREU DE VENDA COMERCIAL	900 €

Es pot veure com el preu del producte final descendeix molt del cost total de producció. Aquest preu s'ha pensat al analitzar la situació econòmica actual del país. S'ha de competir amb les altres empreses que ofereixen el mateix producte. Una aplicació multimèdia personalitzada, com el cas del nostre producte, pot costar entre 1000-1500 o 2000€ en el mercat actual segons el grau de professionalitat.

L'estratègia comercial comença per oferir productes d'alta qualitat i a preu reduït. El més important és oferir quelcom nou o poc vist, dissenys exclusius i formes d'interacció innovadores. Si s'aconsegueix, et pots donar a conèixer i així tirar endavant amb el projecte.

Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Graduat en Mitjans Audiovisuals

PORTAFOLI DIGITAL MULTIMÈDIA

Annexos

JOSEP BORRÀS YÁÑEZ
PONENT: LINA JUAN

TARDOR/PRIMAVERA ANY



TecnoCampus
Mataró-Maresme

Índex.

Annex I. Codi de programació <i>actionsript</i>	1
Annex II. Contingut del DVD.....	17

Annex I. Codi de programació *actionscript*.

Scrollbar:

```
var sd:Number;
var sr:Number;
var cd:Number;
var cr:Number;
var new_y:Number;
var drag_area:Rectangle;

var pf_content:MovieClip;
var pf_content_area:MovieClip;
var scroll_vel:Number; // 0.00 to 1.00

function scrolling( ct:String, ct_area:String, speed:Number ):void
{
    scroll_vel = speed;
    if( scroll_vel < 0 || scroll_vel > 1 ) scroll_vel = 0.50;

    pf_content = parent[ct];
    pf_content_area = parent[ct_area];

    pf_content.mask = pf_content_area;
    pf_content.x = pf_content_area.x;
    pf_content.y = pf_content_area.y;

    pf_scroller.x = pf_scrollable_area.x;
    pf_scroller.y = pf_scrollable_area.y;

    sr = pf_content_area.height / pf_content.height;
    pf_scroller.height = pf_scrollable_area.height * sr;

    sd = pf_scrollable_area.height - pf_scroller.height;
    cd = pf_content.height - pf_content_area.height;
    cr = cd / sd * 1.01;

    drag_area = new Rectangle(0, 0, 0, pf_scrollable_area.height - pf_scroller.height);

    if ( pf_content.height <= pf_content_area.height )
    {
        pf_scroller.visible = pf_scrollable_area.visible = false;
    }

    pf_scroller.addEventListener( MouseEvent.MOUSE_DOWN, scroller_drag );
    pf_scroller.addEventListener( MouseEvent.MOUSE_UP, scroller_drop );
    this.addEventListener( Event.ENTER_FRAME, on_scroll );
}
```

```
function scroller_drag( me:MouseEvent ):void
{
    me.target.startDrag(false, drag_area);
    stage.addEventListener(MouseEvent.MOUSE_UP, up);
}
function scroller_drop( me:MouseEvent ):void
{
    me.target.stopDrag();
    stage.removeEventListener(MouseEvent.MOUSE_UP, up);
}
function up( me:MouseEvent ):void
{
    pf_scroller.stopDrag();
}
function on_scroll( e:Event ):void
{
    new_y = pf_content_area.y + pf_scrollable_area.y * cr - pf_scroller.y * cr;
    pf_content.y += ( new_y - pf_content.y ) * scroll_vel;
}
```

Reproductor de música:

```
/*
Reference
http://www.gotoandlearn.com
*/
stop();
var auto_play:Boolean = true;
var i:Number;
var total_items:Number;
var current_no:Number = 0;
var flashmo_xml:XML;
var flashmo_item_list = new Array();
```

```
var flashmo_s:Sound = new Sound();
var flashmo_sc:SoundChannel = new SoundChannel();
var flashmo_st:SoundTransform = new SoundTransform();
var flashmo_cpi:Number = 0;
var flashmo_lpi:Number = -1;
var flashmo_pos:Number = 0;
var item_paused:Boolean = false;
var playback_min:uint;
var playback_sec:uint;
var total_min:uint;
var total_sec:uint;
var estimated_length:int;

var drag_area:Rectangle = new Rectangle( 0, status_bar.status_bar_scroller.y,
status_bar.load_progress_bar.width, 0 );

status_bar.visible = false;

flashmo_volume.visible = false;

flashmo_previous.visible = false;

flashmo_next.visible = false;

flashmo_play.visible = false;

flashmo_pause.visible = false;

current_item_title.text = "Loading XML Data...";

status_bar.status_bar_scroller.buttonMode = true;

status_bar.status_bar_scroller.addEventListener( MouseEvent.MOUSE_DOWN,
scroller_drag );

status_bar.status_bar_scroller.addEventListener( MouseEvent.MOUSE_UP, scroller_drop
);
```

```
status_bar.fm_bar.addEventListener( Event.ENTER_FRAME, item_progress );

status_bar.clickable_area.buttonMode = true;

status_bar.clickable_area.addEventListener( MouseEvent.MOUSE_DOWN,
position_seek );

status_bar.load_progress_bar.addEventListener( Event.ENTER_FRAME,
drag_area_progress );

function load_xml( xml_file:String ):void{

    var xml_loader:URLLoader = new URLLoader();

    xml_loader.load( new URLRequest( xml_file ) );

    xml_loader.addEventListener(Event.COMPLETE, read_xml);

}

function read_xml( e:Event ):void {

    flashmo_xml = new XML(e.target.data);

    total_items = flashmo_xml.song.length();

    for( i = 0; i < total_items; i++ )

    {

        flashmo_item_list.push( {

            url: flashmo_xml.song[i].url.toString(),

            artist: flashmo_xml.song[i].artist.toString(),

            track: flashmo_xml.song[i].track.toString()

        } );

    }

}
```

```
    }  
  
    play_item();  
  
    add_controls();  
}  
  
function add_controls():void{  
  
    flashmo_previous.addEventListener(MouseEvent.CLICK, on_click_prev);  
  
    flashmo_pause.addEventListener(MouseEvent.CLICK, on_click_pause);  
  
    flashmo_play.addEventListener(MouseEvent.CLICK, on_click_play);  
  
    flashmo_next.addEventListener(MouseEvent.CLICK, on_click_next);  
  
    status_bar.visible = true;  
  
    flashmo_volume.visible = true;  
  
    flashmo_previous.visible = true;  
  
    flashmo_next.visible = true;  
  
    if( auto_play )  
    {  
  
        flashmo_pause.visible = true;  
  
        flashmo_play.visible = false;  
  
    }  
  
    else
```

```
{  
  
    flashmo_play.visible = true;  
  
    flashmo_pause.visible = false;  
  
}  
}  
  
function play_item():void  
{  
  
    status_bar.load_progress_bar.removeEventListener(Event.ENTER_FRAME,  
drag_area_progress);  
  
    flashmo_s.removeEventListener(ProgressEvent.PROGRESS, load_progress);  
  
    status_bar.load_progress_bar.scaleX = status_bar.clickable_area.scaleX =  
  
    status_bar.fm_bar.width = status_bar.status_bar_scroller.x = 0;  
  
    flashmo_sc.stop();  
  
    flashmo_s = new Sound();  
  
    flashmo_sc = new SoundChannel();  
  
    flashmo_volume.mute.gotoAndStop("on");  
  
    if( flashmo_volume.flashmo_scroller.x == 0 )  
  
        flashmo_volume.flashmo_scroller.x = 60;  
  
    flashmo_st.volume = flashmo_volume.flashmo_scroller.x /
```



```
( flashmo_volume.flashmo_scrollable_area.width-
flashmo_volume.flashmo_scroller.width );

    flashmo_s.load( new URLRequest( flashmo_item_list[flashmo_cpi].url ) );

status_bar.load_progress_bar.addEventListener( Event.ENTER_FRAME,
drag_area_progress );

    if( flashmo_cpi != 0 || auto_play ){

        flashmo_sc = flashmo_s.play();

        item_paused = false;

        flashmo_pause.visible = true;

        flashmo_play.visible = false;

    }

    else

    {

        flashmo_pause.visible = false;

        flashmo_play.visible = true;

    }

flashmo_sc.soundTransform = flashmo_st;

flashmo_sc.addEventListener(Event.SOUND_COMPLETE, next_item_handler);

flashmo_s.addEventListener(ProgressEvent.PROGRESS, load_progress);

    flashmo_lpi = flashmo_cpi;
```

```
        current_item_text.text = (flashmo_cpi+1) + " of " + total_items;

current_item_title.text = flashmo_item_list[flashmo_cpi].artist
+ " - " + flashmo_item_list[flashmo_cpi].track;
}

function next_item_handler(e:Event){

    play_next_item();

}

function load_progress(e:ProgressEvent):void

{

var load_ratio:Number = e.bytesLoaded / e.bytesTotal;

status_bar.load_progress_bar.scaleX = status_bar.clickable_area.scaleX = load_ratio;

    if( load_ratio >= 1 )

    {

flashmo_s.removeEventListener( Event.ENTER_FRAME, load_progress );

    }

}

function on_click_pause( me:MouseEvent ):void

{

    flashmo_play.visible = true;

    flashmo_pause.visible = false;
```

```
        pause_it();

    }

    function on_click_play( me:MouseEvent ):void
    {

        flashmo_pause.visible = true;

        flashmo_play.visible = false;

        unpause_it();

    }

    function pause_it():void
    {

        item_paused = true;

        flashmo_pos = flashmo_sc.position;

        flashmo_sc.stop();

        flashmo_sc.removeEventListener(Event.SOUND_COMPLETE, next_item_handler);

    }

    function unpause_it():void
    {

        item_paused = false;

        flashmo_sc = flashmo_s.play(flashmo_pos);

        flashmo_sc.addEventListener(Event.SOUND_COMPLETE, next_item_handler);

    }

}
```

```
}  
  
function on_click_next( me:MouseEvent ):void{  
    play_next_item();  
}  
  
function on_click_prev( me:MouseEvent ):void{  
    play_prev_item();  
}  
  
function play_next_item():void  
{  
    if( flashmo_cpi < total_items - 1 )  
        flashmo_cpi++;  
    else  
        flashmo_cpi = 0;  
  
    play_item();  
}  
  
function play_prev_item():void  
{  
    if(flashmo_cpi > 0)  
        flashmo_cpi--;
```

```
else

    flashmo_cpi = total_items - 1;

    play_item();
}

function scroller_drag( me:MouseEvent ):void{

    me.target.startDrag(false, drag_area);

    status_bar.status_bar_scroller.addEventListener( Event.ENTER_FRAME,
on_scroll );

    status_bar.fm_bar.removeEventListener( Event.ENTER_FRAME, item_progress
);

    stage.addEventListener( MouseEvent.MOUSE_UP, up );
}

function scroller_drop( me:MouseEvent ):void{

    me.target.stopDrag();
}

function position_seek( me:MouseEvent ):void{

status_bar.status_bar_scroller.startDrag(true, drag_area);

status_bar.status_bar_scroller.addEventListener( Event.ENTER_FRAME, on_scroll );

status_bar.fm_bar.removeEventListener( Event.ENTER_FRAME, item_progress );

stage.addEventListener( MouseEvent.MOUSE_UP, up );
```

```
}  
  
function up( me:MouseEvent ):void{  
  
    status_bar.status_bar_scroller.stopDrag();  
  
    set_progress();  
  
    status_bar.fm_bar.addEventListener( Event.ENTER_FRAME, item_progress );  
  
    status_bar.status_bar_scroller.removeEventListener( Event.ENTER_FRAME,  
on_scroll );  
  
    stage.removeEventListener( MouseEvent.MOUSE_UP, up );  
  
}  
  
function on_scroll( e:Event ):void{  
  
status_bar.fm_bar.width = status_bar.status_bar_scroller.x;  
  
var playback_ratio:Number =  
(status_bar.fm_bar.width/status_bar.load_progress_bar_bg.width )  
  
        * estimated_length * 0.001;  
  
    playback_min = Math.floor(playback_ratio / 60);  
  
    playback_sec = Math.floor(playback_ratio % 60);  
  
  
status_bar.status_text.text = pad_zero(playback_min) + ":" +pad_zero(playback_sec)+ " /  
" + pad_zero(total_min) + ":" + pad_zero(total_sec);  
  
}  
  
function set_progress():void{  
  
var load_time:Number = flashmo_s.bytesLoaded / flashmo_s.bytesTotal;
```

```
var estimated_length:int = Math.ceil( flashmo_s.length / load_time );

var pos:Number = estimated_length * ( status_bar.status_bar_scroller.x /
status_bar.load_progress_bar_bg.width );

    if( !item_paused ){

        if( Math.ceil(pos) == estimated_length ){

            play_next_item();

        }

        else

        {

            flashmo_sc.stop();

            flashmo_sc = flashmo_s.play(pos);

flashmo_sc.addEventListener( Event.SOUND_COMPLETE, next_item_handler );

        }

    }

    else

    {

        if( pos == estimated_length )

            pos = pos - 1;

        flashmo_pos = pos;

    }

}
```

```
function item_progress( e:Event ):void
{
    if( !item_paused )
    {
        var load_time:Number = flashmo_s.bytesLoaded / flashmo_s.bytesTotal;

        estimated_length = Math.ceil( flashmo_s.length / load_time );

        playback_min = Math.floor( ( flashmo_sc.position * 0.001 ) / 60);
        playback_sec = Math.floor( ( flashmo_sc.position * 0.001 ) % 60);

        total_min = Math.floor( ( estimated_length * 0.001 ) / 60 );
        total_sec = Math.floor( ( estimated_length * 0.001 ) % 60 );

        status_bar.fm_bar.width = Math.round( flashmo_sc.position /
flashmo_s.length * status_bar.load_progress_bar.width );

        status_bar.status_bar_scroller.x = status_bar.fm_bar.width;

        status_bar.status_text.text = pad_zero(playback_min) + ":" +
pad_zero(playback_sec) + " / " + pad_zero(total_min) + ":" + pad_zero(total_sec) ;

    }
}

function drag_area_progress( e:Event ):void{

    drag_area = new Rectangle(0, status_bar.status_bar_scroller.y,
status_bar.load_progress_bar.width, 0);
```



```
        if( status_bar.load_progress_bar.width == status_bar.load_progress_bar_bg.width
    )
        status_bar.load_progress_bar.removeEventListener(
Event.ENTER_FRAME, drag_area_progress );
    }
function pad_zero( str_to_pad:Number ):String{
    if(str_to_pad < 10)
        return "0" + str_to_pad;
    else
        return String( str_to_pad );
}
```


Annex III. Contingut del DVD.

- Documentació del projecte (memòria, estudi econòmic i annexes).
- Arxius *.fla* i *.swf* de l'aplicació
- Tots els *scripts* necessaris (*html, as, xml, css, js*).
- Arxius de vídeo, imatges i música.