

UNIVERSITAT OBERTA DE CATALUNYA

Tim & Timéa

-

Portatge d'una aplicació multimèdia per a l'aprenentatge electrònic (e-learning)
del francès oral bàsic, de iOS cap a Android mitjançant la utilització de HTML5

per

Stiepan Aurélien Kovac

Memòria tècnica corresponent al treball de fi de carrera de:
Enginyeria Tècnica de Telecomunicacions,
especialitat de Telemàtica

en els

Estudis d'Informàtica, Multimèdia i Telecomunicació

3 de febrer de 2014

Per als Nicolas, Jaro i Dida

Abstract

Estudis d'Informàtica, Multimèdia i Telecomunicació

Memòria tècnica corresponent al TFC d'ETTT

per Stiepan Aurélien Kovac

Paraules clau: aplicació mòbil, HTML5, Android, iOS, e-learning, portabilitat, Tim & Timéa FR, aprenentatge d'idiomes, TTS, Phonegap, Cordova

Català:

La finalitat del present Treball Final de Carrera (TFC) és adaptar l'aplicació per a iPhone Tim & Timéa FR a la plataforma Android. Concretament, el programari original, escrit en C++ utilitzant OpenGL ES¹ 2.0 i llibreries pròpies a iOS¹ (versió 4 i +), s'adaptarà a Android 4.x, resolució 720p sota la forma d'un programari HTML5¹, el nou estàndard per al desenvolupament d'aplicacions multimèdia a Internet (font: <http://www.w3.org/TR/2013/WD-html51-20131029/> en anglès), contingut dintre d'un contenidor Android WebView, utilitzant la tecnologia provada Phonegap/Cordova per a fer-ho i proveir algunes funcions que HTML5 encara no proveïx satisfactòriament, tal com l'integració de vídeos fora de línia a l'aplicació (en el context d'una aplicació integrada a un contenidor WebView), o bé el TTS, que no està implementat a navegadors mòbils, tot i que l'estàndard ho preveu.

El treball consisteix doncs en recuperar els recursos multimèdia - sons, imatges i animacions/vídeos - de l'aplicació per a iPhone i adaptar-los a Android (quan cal) per una banda, i programar la part funcional en HTML5 des de zero, imitant el funcionament de la versió iPhone el més fidelment possible, per l'altra banda; l'eina d'implementació principal serà Construct 2. El fet d'utilitzar HTML5 i doncs, de programar la part funcional des de zero, es motiva de manera detallada ulteriorment, en la part d'Estat de l'Art. No obstant, s'ha de destacar aquí el fet que l'ús de HTML5 permet, almenys teòricament, una compatibilitat amb tots els sistemes operatius mòbils que suporten aquest estàndard, entre altres el Windows 8, sense que el codi s'hagi de canviar de manera significativa a l'hora de portar* (entendre *port* en anglès) l'aplicació cap a d'altres sistemes operatius.

Finalment, per a verificar el postulat de compatibilitat amb tots els SO mòbils que suporten HTML5, es procedeix a un *backport* (portatge al revers) parcial de l'aplicació

HTML5 cap a la plataforma[†] de Apple que consisteix el iPad amb el seu sistema operatiu, iOS.

*Neologisme per a denotar el fet d'adaptar un programari existent a un altre SO¹ .

†Per plataforma, s'entén la combinació entre material, com ara un PC o tauleta, i SO.

¹Veure la pàgina [viii](#) per una llista de les abreviacions emprades i el significat associat; estan referenciades només la primera vegada que apareixen.

Castellano:

Se trata de adaptar la aplicación para iPhone Tim & Timéa FR a la plataforma Android. Concretamente, el programa original, escrito en C++ utilizando OpenGL ES 2.0 i librerías propias a iOS (versión 4 i +), se adaptará a Android 4.x, resolución 720p, bajo la forma de una aplicación HTML5, el nuevo estándar para el desarrollo de aplicaciones multimedia al Internet (fuente: <http://www.w3.org> en inglés), contenido en un *container* Android WebView, utilizando la tecnología probada Phonegap/Cordova para hacerlo y proveer algunas funciones que HTML5 aún no provee satisfactoriamente, como la integración de vídeos fuera de línea a la aplicación (dentro del contexto de una aplicación integrada a un contenedor WebView), u el TTS, que no esta implementado en navegadores móviles, aunque el estándar lo prevé.

El trabajo consiste por tanto en recuperar los recursos multimedia - sonidos, imágenes y vídeos / animaciones - de la aplicación para iPhone y adaptarlos a Android (cuando es necesario) por un lado, y programar la parte funcional en HTML5 des de cero, imitando el funcionamiento de la versión iPhone lo más fielmente posible, por el otro lado; la herramienta de implementación principal será Construct 2. El hecho de utilizar HTML5 y por tanto, de programar la parte funcional des de cero, se motiva de manera detallada ulteriormente, en la parte de Estado del Arte. No obstante, se ha de destacar aquí el hecho que utilizar HTML5 permite, por lo menos teóricamente, una compatibilidad con todos los sistemas operativos móviles que suporten este estándar, entre otros Windows 8, sin que el código se haya de cambiar de forma significativa a la hora de portear* (entender *port* en inglés) la aplicación a otros SO

Finalmente, para verificar el postulado de compatibilidad con todos los SO móviles que suporten HTML5, se procede a un *backport* (portaje al revés) parcial de la aplicación HTML5 a destinación de la plataforma[†] de Apple que consisteix el iPad y su sistema operativo, iOS.

*Neologismo para denotar el hecho de adaptar un programa existente a un otro SO.

†Por plataforma, se entiende la combinación de material, como un PC o tableta, y SO.

Agraïments

Voldria demostrar el meu màxim agraïment a totes les persones que van fer possible aquest treball i els meus estudis a la UOC en general, en particular els meus pares i avis, la família d'Agraives, el projecte Apertium que em va permetre escriure en català des del inici dels meus estudis a la UOC, la Sra. Marielle Deriaz per a la seva gentil autorització d'utilitzar els recursos del joc Tim & Timéa per a iOS en aquest TFC i tot l'equip darrere el videojoc, especialment la Sra. Milena Chernysheva i el Sr. Grigori Chevtchenko per als dibuixos i animacions, respectivament el motor gràfic OpenGL a versions Apple del videojoc, que van servir com a base i referència per al present treball de portatge utilitzant HTML5, així com les màximes gràcies al Sr. Ferran Domínguez Gros i al Sr. David García Solórzano, que van possibilitar la realització del TFC durant el present semestre malgrat dificultats administratives encontrades a l'inici, respectivament guiar-me durant l'elaboració d'aquest treball.

Índex

Abstract	ii
Agraïments	iv
Índex de Taules	vii
Abreviacions	viii
1 Introducció	1
1.1 Objectius principals	1
1.2 Beneficis	1
1.3 Productes obtinguts	2
1.3.1 Memòria	2
1.3.2 Presentació i demostració	2
1.3.3 Programes	3
1.4 Motivació	4
1.4.1 Portabilitat de les aplicacions per a e-learning	4
1.4.2 Tecnologies del llenguatge	4
2 Estat de l'Art	5
2.1 Introducció al <i>e-learning</i> multimèdia	5
2.2 Entorns i programari històrics	6
2.2.1 Al PC/Mac: Authorware	6
2.2.2 Al Internet: els <i>plug-ins</i>	7
2.2.3 Als mòbils: J2ME	7
2.3 Tecnologies contemporànies	8
2.3.1 Web 2.0(.1): des de Ajax cap a aplicacions HTML5	9
2.3.2 Tecnologia nativa: el cas d'OpenGL (ES)	10
2.3.3 Tècnica híbrida: Phonegap/Cordova	11
2.4 Aplicacions m-learning contemporànies	11
2.4.1 Aplicacions m-learning generalistes	11
2.4.2 Aplicacions m-learning especialitzades	12
2.4.3 Aplicacions m-learning per a nens	12
2.4.3.1 Aplicacions lúdiques	12
2.4.3.2 Els <i>Serious games</i>	13
2.4.3.3 Ètica i negoci	13

3	Disseny i implementació	15
3.1	Del portatge d'aplicacions i de la portabilitat	15
3.2	Entorn de desenvolupament	16
3.3	Estructura de l'aplicació	18
3.3.1	Estructura original	18
3.3.2	Estructura modificada	19
3.3.3	Estructura final	19
3.4	Portatge d'aplicacions iOS cap a Android	22
3.4.1	Solucions sense HTML5	22
3.4.1.1	Solucions automàtiques: Apportable	22
3.4.1.2	Solucions manuals	22
3.4.2	Solucions utilitzant HTML5	23
3.4.2.1	Solucions automàtiques: Intel XDK	23
3.4.2.2	Solucions manuals	23
3.4.2.3	Limitacions de HTML5 - solucions semi-natives	23
3.5	HTML5 al mòbil: estàndard(s) vs implementació	24
3.5.1	L'objecte àudio	24
3.5.2	L'objecte vídeo	24
3.5.3	Formats de pantalla	25
3.5.4	Implementació dels dibuixos animats	26
3.5.4.1	En general	26
3.5.4.2	Al Construct 2	26
3.6	Generació vocal per mitjà de programari al entorn mòbil	27
3.6.1	Generació preparada	27
3.6.2	Generació a l'execució	28
3.6.3	Implementació en complement al àudio HTML5	28
4	Proves i experiments	30
4.1	Proves	30
4.1.1	Proves funcionals individuals	30
4.1.2	Proves funcionals globals	33
4.2	Experiments	33
5	Conclusions i línies de futur	36
5.1	Conclusions	36
5.2	Línies de futur	36
5.2.1	Per al m-learning utilitzant HTML5	36
5.2.2	Concernint l'aplicació desenvolupada	37
	Bibliografia	38

Índex de taules

3.1	Estructura original	18
3.2	Estructura modificada	19
3.3	Estructura final	20
3.4	Estructura actual	21

Abreviacions

OpenGL (ES)	O pen G raphics L ibrary (for E mbedded S ystems)
iOS	i Phone O perating S ystem, utilitzat també als i Pad/ P od (T ouch)/...
HTML5	H ypertext T ext M arkup L anguage, versió 5 [1] (<i>5.1 des del 29.10.2013</i>)
SO	S istema O peratiu, de O perating S ystem en anglès
FPS	F rames P er S econd
IPS	I matges P er S egon
apk	A ndroid a pplication p ackage file
NFP	N atural F amily P lanning
UDID	U nique D evice I Dentifier
AJAX	A synchronous J ava S cript + X ML (<i>o qualsevol solució javascript equivalent</i>)
OMS	O rganització M undial de la S anitat
GSM	G lobal S ystem for M obile telecommunications
VM	V irtual M achine, anglès per a M aquina V irtual
js	j avascript, el llenguatge de script utilitzat per a H TML
URI	U niform R esource I dentifiers, definit per la RFC 2396
PNG	P ortable N etwork G raphics
FLA	F lash M ovie A uthoring file
API	A pplication P rogramming I nterface

Capítol 1

Introducció

1.1 Objectius principals

L'objectiu principal del TFC és la conversió de l'app Tim & Timéa de iOS a Android. Fruit d'aquest procés, també es vol verificar la factibilitat i els beneficis del portatge de una aplicació per a m-learning* programada d'una manera clàssica (aquí, C++/OpenGL ES) cap a d'altres plataformes utilitzant HTML5, tot i proposant una metodologia bastant genèrica per a poder adaptar-la a uns altres casos similars de portatge d'aplicacions multimèdia programades de manera clàssica cap a noves plataformes, sempre mitjançant l'ús de HTML5.

*E-learning, o aprenentatge electrònic, mitjançant dispositius mòbils com smartphones, tauletes, etc. Serà aprofundit en l'estat de l'Art (Capítol 2).

1.2 Beneficis

Poder portar aplicacions per a e-learning/m-learning destinades originalment a una sola plataforma determinada a un ventall ampli de plataformes, utilitzant una metodologia genèrica que reutilitza els recursos multimèdia, ho que sol implicar un guany important en temps com en diner.

1.3 Productes obtinguts

En aquest apartat es presenten breument els productes obtinguts durant l'elaboració del present TFC; els detalls tècnics per a l'elaboració dels productes “accessoris” que són la memòria i la demostració vídeo del resultat principal - l'aplicació Tim & Timéa per a Android 4+ (i la seva elaboració, la que serà demostrada mitjançant un *screencast* al document de presentació) - figuren als annexos.

1.3.1 Memòria

Aquest document s'ha escrit amb l'eina \LaTeX , més precisament la distribució \MacTeX per a Mac OS X, tal com el seu nom ho pautava, més l'eina gràfica \TeXShop que s'ha utilitzat en concret per a editar els fitxers `.tex*` i l'aplicació BibDesk, també gràfica, per a editar la bibliografia.

Com a base, s'ha pres el model de tesis de doctorat del Dr. Sunil Patel. Tot i que no es tracti d'un doctorat en el present cas, s'ha considerat adient l'ús de \LaTeX i del model de tesis triat en particular, per a la seva facultat de generar la majoria de l'estructura i de la forma del document, deixant l'autor concentrar-se en la seva redacció.

*Els detalls tècnics per a la configuració de l'entorn en vista de la redacció en català, incloent la integració d'un corrector ortogràfic català extern al \TeXShop , s'especifiquen en els annexos.

1.3.2 Presentació i demostració

El vídeo de demostració *in game* de l'aplicació s'ha capturat utilitzant l'eina screenrecord integrada a la versió 4.4 (*code name* Kit Kat) del SO Android, que ens ha permès capturar vídeos amb una taxa d'imatges per segon aproximant 15 FPS (considerem que $\text{IPS} = \text{FPS}^1$), ho que s'ha considerat com a acceptable en el cas de la present aplicació, on no hi ha cap escena amb molts canvis d'imatges ràpids: el vídeo de l'aniversari de Kelly és el cas més extrem (sí, ara es pot capturar el vídeo d'un vídeo amb Android!), on quedem però al límit de 15 FPS.

Tot i així, el maquinari utilitzat (un Galaxy S3) sent una miqueta lent respecte al les especificacions dels models que venen preinstal·lats amb Android 4.4, tal com el Nexus

5 de Google (fabricat per LG), hi ha un lag perceptible entre la vídeo i el só, que per la seva part s'ha capturat passant per un cable mini-jack, el SO Android no tenint una eina comparable al screenrecord per al só: només existeix una eina per a obtenir la forma aproximada de l'ona sonora, amb aquesta vegada una pèrdua no acceptable de qualitat. Com s'ha pogut obtenir un Nexus 5 poc abans del final del TFC, l'aplicació no s'ha optimitzat per a aquest model (que presenta algun *bug* gràfic** respecte al S3 amb Kit Kat...). No obstant, s'ha generat un segon vídeo sense optimitzacions específiques per a aquest model, és a dir utilitzant la mateixa versió de Tim & Timéa per a Android que per als altres models i s'ha verificat que amb un maquinari adaptat, el *lag* present quan el vídeo es pren amb el Galaxy S3, ara està menys present†, tot i que es pot veure encara, sobretot en el cas d'animacions construïts com antics dibuixos animats (imatge per imatge), on hi ha a vegades més de 15 FPS, la més simptomàtica sent la del conill en l'habitació n° 1 del joc Tim & Timéa, on tenim gairebé un segon de retard respecte al cas on no es captura la vídeo.

Finalment, l'àudio capturat de manera analògica amb el programari Audacity per a Mac OS X (també disponible a d'altres SO), s'ha combinat amb la vídeo utilitzant Mp4box.

**El *bug* en qüestió està present al vídeo pres amb el Nexus; s'en mostra una solució a la pàgina 23 del document *kstiepan_tfc_annexos.pdf*.

†Aquest vídeo (sense àudio), el vídeo pres amb el S3 (amb só) així com instruccions per a l'ús de screenrecord, Audacity (incloent instruccions per a filtrar el soroll després de la captura) i Mp4box estan als annexos.

1.3.3 Programes

L'aplicació Tim & Timéa per a Android, sota la forma d'un fitxer apk¹ instal·lable a qualsevol dispositiu Android a partir de la versió 4.0 i d'una resolució *HD-ready*, és a dir (1280x)720p. Conté les escenes i mini-jocs de base del joc original amb adaptacions menors, sense però totes les variacions d'escenes idèntiques amb objectes i veus diferents, que s'han deixat per a una fase d'implementació ulterior en vista d'una publicació al Google Play. A més a més, en els propers capítols, es justifica per què i s'explica com s'ha procedit amb l'eina Construct 2 i altres eines complementaries per a efectuar el *port* d'iOS a Android mitjançant l'utilització de HTML5 i l'integració de l'aplicació HTML5 a un contenidor Cordova (ex-Phonegap) 2.9.1 per a Android.

¹Veure la pàgina [viii](#) per una llista de les abreviacions emprades i el significat associat.

1.4 Motivació

1.4.1 Portabilitat de les aplicacions per a e-learning

Interès per a l'aprenentatge per mitjà de recursos electrònics, per a les tecnologies que ho habiliten i en particular per al HTML5, tecnologia que sembla per fi aportar ho que d'altres tecnologies tal com Java, solien aportar - és a dir, una abstracció de la plataforma sobre la qual el programari s'executa - però no han assolit del tot, almenys no a un nivell tant perceptible per l'usuari final que amb HTML5.

1.4.2 Tecnologies del llenguatge

Interès per a les tecnologies del llenguatge (informàtica lingüística, etc.), els *serious games*, que són (vídeo-)jocs amb finalitats serioses, aquí d'aprenentatge: no obstant la simplicitat aparent de l'aplicació original, integra conceptes avançats de psicologia del aprenentatge*; l'escenari va ser dissenyat per especialistes del camp, incloent logopedes i psicòlogues especialitzades en eines electròniques per a l'aprenentatge de la llengua (el francès, en l'ocurrència).

*fer clic [aquí](#) o [aquí](#) per a un article detallat de la Viquipèdia en castellà sobre el tema, respectivament el programa de Màster en disseny de *serious games* que ofereix l'universitat d'estat del Michigan (EUA), per a una síntesi de l'ampli ventall de subjectes relacionats amb el concepte en qüestió.

Capítol 2

Estat de l'Art

2.1 Introducció al *e-learning* multimèdia

Quan estem parlant de *e-learning* avui dia, implica sovint elements multimèdia, tot i que això encara no està la norma, sobretot en el cas de l'estudi asíncron de branques matemàtiques o d'altres subjectes molt formals, on el paper (sigui electrònic o no) continua essent la forma de predilecció com a base per a l'aprenentatge.

Per *e-learning* asíncron, s'entén estudis per correu electrònic i/o correspondència per Internet d'una altra forma, com es passa a la UOC; per oposició a l'aprenentatge el síncron, que es fa en directe, sovint interaccionant amb altra gent per mitjà d'un xat o similar, tot i que no ha de ser així: també existeix aprenentatge electrònic síncron, que no implica d'altres persones en el moment de la seva utilització. En aquests casos, el procés per a l'aprenentatge sol ser encara més interactiu per tal de compensar la manca d'interactivitat humana.

El *e-learning* multimèdia, per tant, s'aplica més volunters a branques de les matemàtiques aplicades (p.e. vídeos per a explicar principis de criptografia) o d'altres subjectes menys formals com l'aprenentatge d'una llengua i s'adapta particularment bé a entorns d'aprenentatge síncrons, que forçadament són interactius - sigui amb el maquinari i/o d'altres persones - per a fer-los més atractius per una banda i ampliar les possibilitats tècniques que ofereixen, per l'altra.

Arribem aquí als videojocs, que són probablement la forma més interactiva d'aplicacions per a ordinadors i/o maquinari similar; de fet, esdevenen la resposta “natural” a la problemàtica de la manca d'interactivitat.

2.2 Entorns i programari històrics

En aquesta secció es presenten els entorns i programaris històrics, a vegades encara utilitzats arreu del món, que van tenir èxit per a l'elaboració d'aplicacions multimèdia per a *e-learning* més o menys interactives.

2.2.1 Al PC/Mac: Authorware

Molt utilitzat en l'àmbit estatal, fins i tot dins de les forces armades dels EUA* i per supòsit, de països que comparteixen la mateixa tecnologia militar. Es un entorn de programació visual basat en el concepte de línia de flux, que permet desenvolupar aplicacions “rich media” per a *e-learning*, segons el que s'explica a la pàgina web d'Adobe dedicada a Authorware.

L'ús de la paraula composta “rich media” no és per casualitat, sinó que denota el fet que permet crear aplicacions amb un cert nivell d'interactivitat, que podríem situar entre el que es feia abans de HTML5 i el que es fa amb ell: per exemple, pel fet de generar aplicacions per a Mac o Windows, inclou suport per al àudio, però si es volen afegir animacions (de tipus film, sense interactivitat amb aquestes), s'ha de recórrer a Flash Pro per tal de crear-les i després afegir-les com a contingut *plug-in*.

Existia també una possibilitat d'integració web amb el Authorware *Player*, que però mai ha funcionat correctament amb els navegadors de l'època (existeixen *workarounds*, com p.e. AWFix).

*Hi havia fins i un *Aviation Military and Government Authorware Conference*, fins al 2006, poc després de la fusió entre l'editor d'Authorware - Macromedia - i Adobe, que va marcar la fi oficial del programari el 2007 (font: Wikipedia).

2.2.2 Al Internet: els *plug-ins*

Abans de l'advent de HTML5, es podien crear pàgines amb interactivitat i contingut multimèdia bastant limitats, si no es feia ús dels anomenats *plug-ins*, que són programaris externs que s'inclouen durant el carregament d'una pàgina web i accedeixen a funcions del sistema operatiu que no es podien utilitzar des de HTML, com p.e. les funcions necessàries per a reproduir vídeo. Per a aquest últim, el *plug-in* més conegut és, com en el cas d'Authorware, Flash i encara s'utilitza molt a webs tant populars com Youtube (que però, també està disponible en HTML5), o bé com a *fallback*, o bé per defecte, el que sovint és el cas.

Per a aplicacions d'*e-learning* interactives, en canvi, s'han utilitzat molt fins fa poc (degut a problemes de seguretat**) les anomenades *applets* Java; un bon exemple per al seu ús en l'àmbit educatiu és la següent pàgina de l'universitat d'Aizu, al Japó: <http://web-ext.u-aizu.ac.jp/niki/courses/sccp/>.

**Aquest subjecte dona lloc a molta controvèrsia: per a alguns experts, Oracle, que va comprar Sun - i doncs Java - el 2009, està intencionalment abandonant el manteniment de la seguretat de les *applets*, mentre que Oracle afirma que ho està "millerant" (veure p.e. <http://www.oracle.com/technetwork/java/javase/tech/java-code-signing-1915323.html>), el problema sent que ho fa sovint trencant funcionalitat existent.

Atès que no es pot fer més aplicacions amb *applets* Java que funcionin fora de línia (font: discussió al fòrum d'experts StackOverflow), Oracle recomana a partir de la versió 6 de Java migrar-les a aplicacions de tipus *Web Start*, que funcionen fora del navegador un cop baixades (que, però, també pateixen del trencament de funcionalitats decidit de manera unilateral per Oracle). Per fi, Oracle sembla haver donat el colp de gràcia a les *applets* integrades a webs - siguin fora de línia o no - pel fet de treure el codi estàndard per a integrar aquestes del seu lloc web - segons aquest article de la Viquipèdia en anglès. (S'hi pot verificar que l'URL a la nota de peu de pàgina n° 25 no funciona)

2.2.3 Als mòbils: J2ME

En el cas de mòbils, abans de l'era dels *smartphones*, o més acuradament, abans de que aquests estiguin popularitzats per l'iPhone, la via principal de portar aplicacions (de qualsevol tipus) a aquests era també basada en Java, concretament la seva versió per a

mòbils, la quina última actualització es diu J2ME (existeixen altres predecessors històrics), que oferia l'avantatge de funcionar amb qualsevol mòbil compatible amb aquesta versió especial de Java. Un exemple d'aplicació J2ME per a *e-learning* era el Mobile Learning Engine de Moodle.

El defecte principal de J2ME és que era en avançada sobre el seu temps, principalment en quant a les infraestructures dels operadors mòbils, que generalment no oferien solucions abordables per al transferiment de dades (veure [2]) i van començar a fer-ho quan ja havia començat la moda dels telèfons "intelligents", que a més ho solucionaven oferint connectivitat WiFi en la majoria dels casos.

J2ME, però, era i és (per als *feature-phones* compatibles) encara una tecnologia robusta, molt bé adaptada per a aplicacions fora de línia, o que poden funcionar sense necessitar accés a una connexió de dades. Un exemple n'és l'aplicació didàctica per a NFP¹ sympto, que existeix en versió J2ME i continua funcionant sense problemes a terminals compatibles, anys després de la seva creació, ho que es podria difícilment garantir (sense *updates* freqüents) en el cas d'aplicacions per a sistemes tal com iOS o Android, on una decisió de canvis per l'empresa que proporciona la plataforma (aquí Apple, o Google) pot fer incompatible tota una sèrie d'aplicacions dissenyades per a versions anteriors del SO i/o maquinari anterior, com ha estat per exemple el cas amb el canvi del *aspect-ratio* del iPhone 5 respecte a versions anteriors, o encara la supressió del suport per al UDID¹ a iOS pel que fa del SO. Això es sent menys marcadament amb Android, tot i que hi hagi una tendència sensible cap a un ecosistema més tancat per la part de Google (p.e. als últims telèfons de referència Nexus, no es pot treure la bateria, ni inserir una targeta SD, en ambdós casos imitant el *modus operandi* d'Apple amb el iPhone).

¹Veure la pàgina [viii](#) per una llista de les abreviacions emprades i el significat associat.

2.3 Tecnologies contemporànies

Les tecnologies contemporànies s'introdueixen aquí, començant per descriure l'evolució de HTML que ha portat cap al HTML5, descripció que serà seguida pel cas concret d'una tecnologia nativa amb gran ubiqüitat: OpenGL i la seva variant ES per als mòbils i altres dispositius de tipus integrat, per acabar amb la descripció dels sistemes híbrids, en concret Cordova, que permeten barrejar HTML5 amb funcions natives encara no

implementades o de manera no suficient pels navegadors integrats* amb els SO mòbils actuals.

*Es interessant notar aquí que el fet que Apple o Google integri el seu navegador per defecte al seu SO mòbil no crea una reacció pública tant forta que amb el cas d'Internet Explorer al Windows! Com estem a un entorn mòbil, "per definició" tancat, no hi ha gairebé cap reacció, tret potser per la part d'alguns desenvolupadors, sobretot en el cas d'iOS on no només Safari *mobile* ve instal·lat per defecte, però a més no es pot reemplaçar per cap programari concurrent (tret els que facin ús d'exactament el mateix motor Webkit en rerefons, com Chrome per a iOS o Dolphin).

Això va donar lloc a la creació de navegadors que es venen com a contenidors per a aplicacions només, evitant així la restricció d'Apple que s'aplica a navegadors venuts com a tals...

La tensió entre sistemes oberts i tancats és doncs al seu màxim per a les tecnologies actuals, amb per una part iniciatives més o menys obertes com HTML5 o Firefox OS i per l'altra, sistemes més o menys tancats d'Apple, Google i Microsoft, que intenten no obstant *surfear* l'un sobre l'altre i sobretot tenir la màxima influència possible sobre la definició dels nous estàndards com HTML5.1.

2.3.1 Web 2.0(.1): des de Ajax cap a aplicacions HTML5

Ajax¹ és un concepte nascut fa una desena d'anys, que ha permès iniciar la transformació de les aplicacions web clàssiques utilitzant el model client-servidor cap a un nou model asíncron, afegint una capa addicional entre el client el servidor, que paradoxalment - a primera vista només - fa que l'aplicació sigui més reactiva, perquè elimina en gran part, o totalment en alguns casos, els temps d'espera causats pel servidor i/o l'estat de la xarxa.

Eines com JQuery (<http://www.jquery.com>) i varies IDEs destinades a Javascript venen afegir una certa facilitat de programació amb Javascript, que és el cement que fa possible l'interacció entre tots els components d'un sistema Ajax i/o HTML5. HTML5, per la seva part, proporciona nous elements interessants per a aplicacions web com el *canvas* i els elements àudio i vídeo, que solen permetre un funcionament sense *plug-ins* d'aplicacions web - solen, perquè depèn de l'implementació als navegadors.

D'aquí és veu immediatament el potencial per a aplicacions destinades a mòbils, que més sovint que maquinari fixos es troben en condicions de cap o feble connexió de dades, per una banda, i no suporten *plug-ins*, almenys no al mateix nivell que dispositius fixos**, per l'altra.

**Alguns mòbils amb Android suporten Flash, tot i que a vegades no tant bé com les versions per a ordinadors, sobretot en el cas d'aplicacions interactives (l'aplicació Flash Love Game n'és un exemple); cap dispositiu amb iOS - tret modificacions no oficials i no a l'abast del gran públic - ho suporta, per decisió "política" d'Apple de no suportar-ho. Pel que fa de Java, tot i que Android es basa en ell en gran part per a la seva capa aplicativa, no suporta *applets* Java. A iOS, el suport oficial de Java és nul.

2.3.2 Tecnologia nativa: el cas d'OpenGL (ES)

OpenGL és un estàndard bastant obert que permet crear aplicacions gràfiques interactives en 2D i/o 3D, utilitzant llenguatges de programació clàssics com C(++), Ada o Java. Al seu torn, OpenGL ES és la seva versió per a dispositius mòbils, de la mateixa manera que J2ME ho era respecte a Java: és una versió reduïda i optimitzada per a aquests entorns. Per això, una aplicació OpenGL és teòricament portable a qualsevol SO contemporani sense gran modificació del codi, però s'introdueix una distinció per a les plataformes mòbils/integrades, que reben un tractament a part. Això no significa que una aplicació OpenGL no es pugui portar cap a una OpenGL ES i viceversa, però depèn de què funcionalitats utilitza l'aplicació en qüestió, que poden ser implementades o no a OpenGL ES.

A més a més, la capa que serveix de contenidor al programari, i en el cas d'una aplicació multimèdia, per donar-li accés a funcions de reproducció àudio del SO, per exemple, difereix de SO en SO, ho que és lògic, car OpenGL es concentra estrictament en la part gràfica de l'aplicació, a més de no dependre d'un llenguatge de programació en concret. Es però aquesta separació perfecta entre la part gràfica i la resta que fa que aplicacions OpenGL multimèdia són només parcialment portables; un exemple d'això és la versió per a Mac OS X de Tim & Timéa, on la part més difícil ha estat de portar tot ho que concerneix àudio de iOS cap a Mac OS X, les funcions del sistema per a la reproducció de só sent estranyament diferents entre els dos SO d'Apple***. Cal doncs considerar si la part gràfica és la part que demana més treball (aquí no és el cas) per a avaluar

la portabilitat d'una aplicació feta amb OpenGL (ES), entre altres factors com el de la diferència entre la versió de base i la ES (aquí aquesta diferència no jugava cap rol).

***Això s'explica probablement pel fet que l'iPhone deu ser basat en el disseny dels iPods pel que fa del àudio, mentre que OS X té un disseny més clàssic, similar al que hi ha a d'altres SO per a PC

2.3.3 Tècnologia híbrida: Phonegap/Cordova

Cordova (ex-Phonegap) és una plataforma de l'Apache *Software Foundation* que permet a aplicacions HTML5 accedir a funcions natives del maquinari on s'executen, incloent-les en contenidors d'aplicacions específics per a cada SO. Està enfocada als mòbils, tal com el seu antic nom ho indica, i permet omplir el *gap* entre l'especificació HTML5 i el seu suport real als navegadors mòbils.

Una altra tecnologia híbrida, que no es detallarà aquí per motius de concisió, és el WebGL, que permet accedir a funcions OpenGL ES 2.0 des del navegador.

2.4 Aplicacions m-learning contemporànies

A continuació, es presenten algunes aplicacions actuals per a m-learning, que és el subconjunt mòbil del *e-learning*, atès que el propòsit del present treball és l'adaptació d'una aplicació mòbil per a *e-learning* envers una altra plataforma, també mòbil.

2.4.1 Aplicacions m-learning generalistes

–Udemy és una aplicació per a Android on es pot comprar i/o fins i tot visualitzar gratuïtament tot tipus de cursos, proposats per experts, cadascú del seu camp; existeix també en versió web per a ordinadors (de fet, la versió per a Android va ser creada posteriorment). Punt negatiu, donat que es vol general: cap possibilitat per a canviar la llengua. Té entre 10'000 i 50'000 instal·lacions, segons Google Play.

–Quipper Quiz és una aplicació per a iOS que té una filosofia més oberta: tothom hi pot afegir continguts (amb les avantatges i inconvenients que això suposa) mitjançant <http://www.quipper.com>. Estan dedicats al m-learning* i ofereixen unes aplicacions

més especialitzades; algunes seran presentades el la propera subsecció. Pel que fa del nombre d'usuaris, Apple no ho publica al iTunes App Store, però donat el nombre de comentaris (1215) podem deduir que probablement en té més que Udemey.

*Tenen també un bloc amb articles molt complets sobre m-learning de *niche*, a començar per l'article referenciat.

2.4.2 Aplicacions m-learning especialitzades

–Gakushu Check és una altra aplicació iPhone de Quipper dedicada al mercat japonès, mentre que l'aplicació multi-plataforma Quipper Courses està dedicada als mercats del sud-est asiàtic que són Thailand, Vietnam, Malaysia i Indonesia. És interessant constatar aquí que la versió iOS gairebé no té usuaris, probablement degut al fet que Android està molt més popular en aquests països: fins ara, els productes Apple fan succés gairebé només als països més rics del món, en raó tant del preu de venda elevat que tenen, que de l'estratègia de màrqueting d'Apple, que visa aquests països en prioritat, sinó exclusivament - a l'Àsia, això significa només Japó i alguns centres urbans importants i rics com Hong-Kong, que però no són representatius de la situació general del país/regió (en el sentit llarg) on es situen.

2.4.3 Aplicacions m-learning per a nens

2.4.3.1 Aplicacions lúdiques

App-kid.fr (<http://app-kid.fr/applications-enfant-android-gratuites>, en francès) proporciona un comparatiu molt comprensiu de les aplicacions gratuïtes que hi ha per a nens petits a Android. Aquest comparatiu s'ha triat perquè l'aplicació Tim & Timéa també està destinada a nens petits i es publicarà com a aplicació gratuïta al Google Play. Es pot veure que no n'hi ha molt, estem doncs encara en un mercat de *niche*, o “at the long tail”, com dirien al bloc de Quipper. Tot i que algunes contenen components tal com un joc de memòria o trencaclosques, només una d'elles es destaca de les altres per la seva compleció, quantitat i qualitat d'activitats educatives, i de fet s'hauria de classificar a l'apartat següent, els *Serious games*.

2.4.3.2 Els *Serious games*

Tenim doncs un primer exemple de joc seriós Noddy Primeros Passos (<http://pi.vu/B2zU> *Oui, Oui, premiers pas* en francès) HD que a més està destinat al mateix públic que Tim & Timéa FR. Té l'avantatge de correspondre amb la sèrie de dibuixos animats del mateix nom, a més de tenir una cadena de televisió estatal francesa (France 5) com a *sponsor*. Un desavantatge és que malgrat el títol, que conté la paraula HD, no està adaptat a tauletes: amb una tauleta 720p (HD-ready) només funciona en orientació vertical i l'imatge està bastant pixelitzada - doncs en cap cas es pot considerar "HD".

Un altre exemple és una més aplicació de Quipper, StraightAce, aquí dedicada a nens que van al *middle-school* als EUA, és a dir nens/joves que estan acabant l'escola obligatòria i entrant a l'adolescència; els exercicis inclosos es concentren en 2 temes, matemàtiques i llengua anglesa (gramàtica + vocabulari) tot i proveint incentius lúdics com la possibilitat de desbloquejar nous fons de pantalla, personatges, etc.

L'únic desavantatge que es pot constatar és que podria ser una mica més interactiu.

Esta disponible per a la plataforma Android, on té de 10'000 a 50'000 instal·lacions, segons Google Play.

2.4.3.3 Ètica i negoci

Un aspecte important dels videojocs educatius és la finalitat que tenen i en què mesura es consideren acceptables èticament, sobretot quan es vol també treure-ne un profit.

En el cas de Tim & Timéa, la qüestió no és tant relacionada amb la finalitat, que els mitjans:

a banda de saber si és recomanable fer un videojoc per a nens petits (risc d'addicció?), caldria considerar també si és oportú donar-los maquinari amb antenes GSM¹, alhora que les radiacions utilitzades per a la transmissió s'han reconegut com a potencialment cancerígenes per l'OMS¹.

Donat aquestes consideracions, a més de sentit comú, sembla recomanable proporcionar-los només maquinari sense antena GSM, p.e. una tauleta sense aquesta opció, o almenys amb el modi avió (fora de línia) activat si es tracta p.e. d'un mòbil dels pares que forçadament en té una. El problema resideix després en la determinació de qui ha d'informar els pares sobre el tema - en el cas d'aplicacions per a Android - Google,

l'operadora, les autoritats sanitàries o l'editor de cada aplicació per a nens? Hi ha per ara un buit jurídic-reglamentari problemàtic a aquest nivell, on ningú no està responsable i doncs ningú no pren la responsabilitat de fer-ho, per por de fer por i perdre clients. Això s'hauria de canviar.

¹Veure la pàgina [viii](#) per una llista de les abreviacions emprades i el significat associat.

Capítol 3

Disseny i implementació

3.1 Del portatge d'aplicacions i de la portabilitat

A partir del moment que coexisteixen dues plataformes (sempre en el sentit de combinació SO/maquinari) concurrents i fonamentalment diferents, i que volem fer funcionar una aplicació que funciona a l'una, a l'altra, necessitem **portar** aquesta aplicació cap a la plataforma de destí.

Depenent de com ha estat codificada, per una banda, i del grau de diferencia entre les plataformes, per l'altra, aquest procés pot anar de trivial a molt complicat i típicament necessitar que es comenci des de zero la programació de l'aplicació en qüestió per a la plataforma de destí.

En el cas de Tim & Timéa, tot i que s'hauria pogut re-utilitzar la part gràfica, és a dir OpenGL ES, del codi, s'ha optat per a codificar tot des de zero amb HTML5, per tal de tenir una fundació que pugui després estar portada cap a qualsevol plataforma suportant HTML5* sense grans esforços, bàsicament canviant el contenidor Cordova emprat per a que correspongui amb la nova plataforma.

*Com p.e. el Windows Phone 8

3.2 Entorn de desenvolupament

Com a entorn de desenvolupament, s'ha configurat una maquina virtual amb Windows 7 per sobre d'un MacBook de 2008 amb un CPU Core 2 Duo @2.1 GHz, 4 Gb de RAM i el SO Mac OS X 10.7.5, a la maquina s'han instal·lat els entorns de desenvolupament Construct 2 (compatible només amb Windows) i Android Studio (instal·lat a la VM¹ Windows per tal de facilitar el flux de treball a l'hora de generar apks).

FIGURA 3.1: Entorn de desenvolupament - vista del *Event sheet* principal de l'aplicació a Construct 2, dins de la maquina virtual

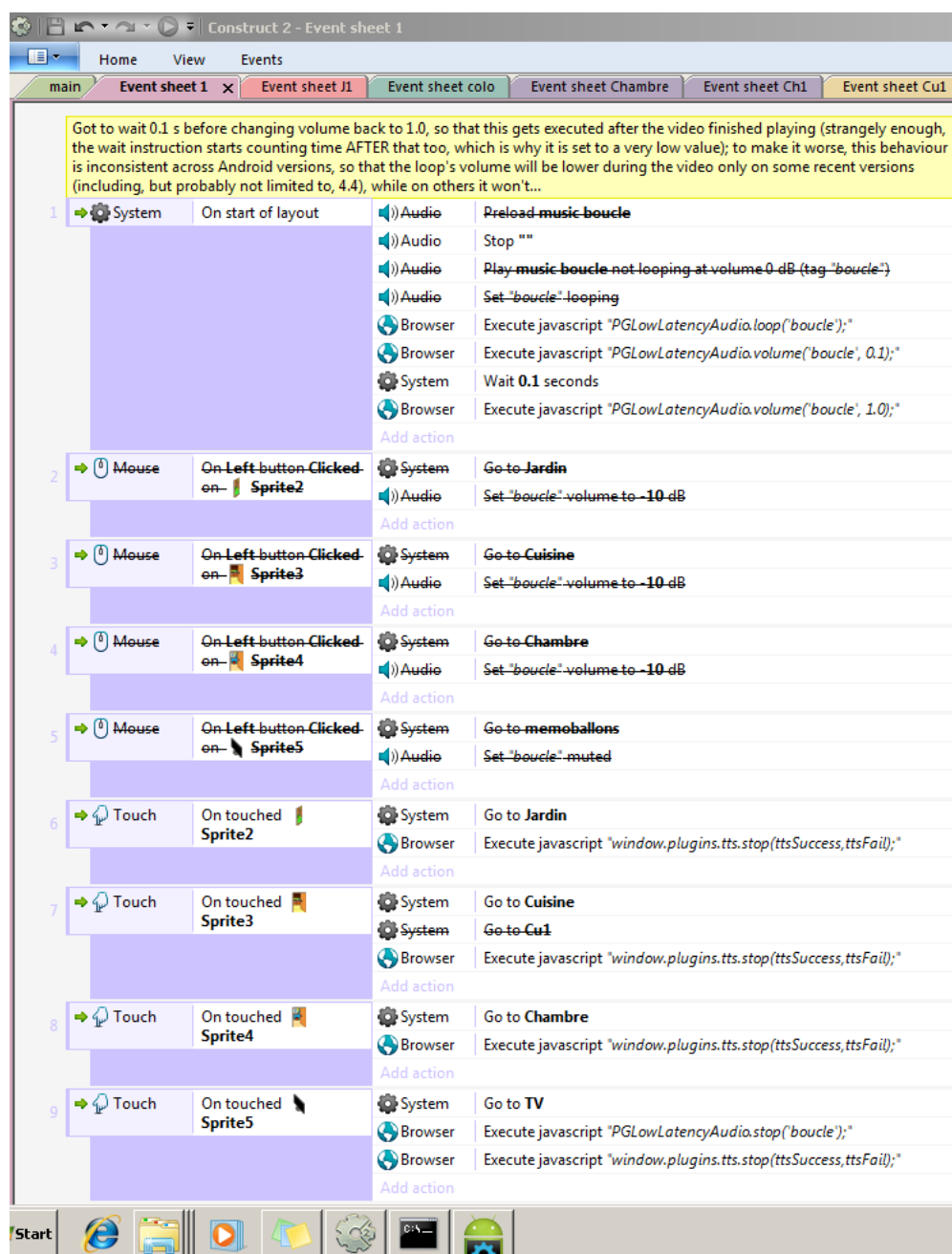
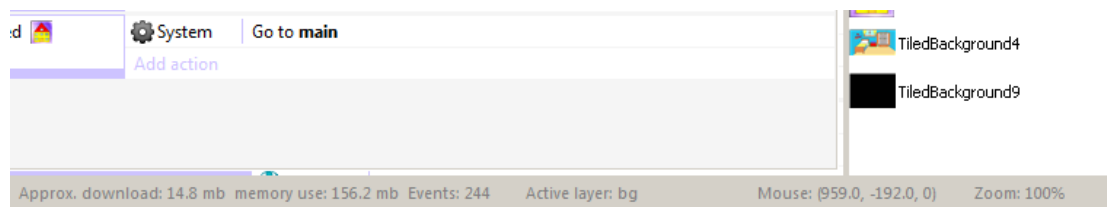


FIGURA 3.2: Zoom al nombre d'esdeveniments; hi ha també una estimació de la mida final de l'aplicació (sense elements externs com vídeos) i del seu consum de memòria



S'ha triat utilitzar el Construct 2 més que una altra eina per al desenvolupament d'aplicacions HTML5 per als següents motius principals:

- Eina gràfica molt pràctica per a definir les escenes del joc i posicionar els objectes, o *sprites*, dins d'aquestes
- Genera codi js¹ ben estructurat, integrable a contenidors Cordova mitjançant modificacions mínimes del fitxer index.html generat
- Possibilitat per a usuaris amb coneixements tècnics bàsics de fer algunes modificacions senzilles, com per exemple canviar alguna veu per una altra -> major genericitat del codi, inclòs per al client

En raó del nombre important d'esdeveniments que comporta l'aplicació, s'ha comprat la versió comercial de Construct 2 per a no tenir limitacions respecte al nombre d'*events*.

En quant al Android Studio*, tot i que està encara a l'estadi d'una versió *early access preview* (ara 0.3.2), s'ha optat per aquesta opció més que Eclipse per a la seva relativa lleugeresa, donat que ja tenim un cert *overhead* degut al fet de treballar amb una VM.

FIGURA 3.3: Vista del *memory overhead* causat per un sistema servidor Windows al sistema de virtualització ESXi: és la memòria que la VM consumeix a més del que consumiria si fos una màquina real. Font: [http://i2.wp.com/jpaul.me/...](http://i2.wp.com/jpaul.me/)

General	
Guest OS:	Microsoft Windows Server 2008 R2 (64-bit)
VM Version:	7
CPU:	1 vCPU
Memory:	4096 MB
Memory Overhead:	183.05 MB
VMware Tools:	Not installed
IP Addresses:	
DNS Name:	
EVC Mode:	N/A
State:	Powered Off
Host:	192.168.126.42
Active Tasks:	

*Per a instruccions sobre com crear un projecte Cordova dins d'aquest IDE, fer clic aquí.

3.3 Estructura de l'aplicació

3.3.1 Estructura original

Heus aquí l'estructura original per al prototip Authorware proposat per la Sra. Deriaz:

TAULA 3.1: Estructura original

dossier choix	dossier sous_menu
2ème planche	3ème planche
Décor : hall d'entrée	
Choix des jeux	Menu intermédiaire
4 zones cliquables pour accéder aux jeux :	Chaque lieu de jeux offre trois jeux différents fonctionnant de la même façon, suivi d'un mini-jeu en récompense
porte du jardin	la télévision offre deux animations à choix suivi d'un mini-jeu
porte de la cuisine	
porte de la chambre	
écran de la télévision	
	jardin_a / coloriage_a
jeu 1 : jardin	jardin_b / coloriage_b jardin_c / coloriage_c
	cuisine_a / memory_a cuisine_b / memory_b cuisine_c / memory_c
jeu 2 : cuisine	
	chambre_a / reparation_a chambre_b / reparation_b chambre_c / reparation_c
jeu 3 : chambre	
	tv_a / puzzle_a tv_b / puzzle_b
jeu 4 : tv	

3.3.2 Estructura modificada

Heus aquí l'estructura modificada, que va servir de base per a la versió iPhone:

TAULA 3.2: Estructura modificada

1ra escena	2na escena	3ra escena
	Decoració : hall d'entrada	
	Tria dels jocs	Menú intermediari
Animació : els 2 nens corren cap a la mare que està davant de la casa i que els invita a entrar per a jugar dins de la casa	4 zones clicables per accedir als jocs : porta del jardí porta de la cuina porta del dormitori pantalla de la TV	Cada lloc/habitació ofereix tres jocs diferents funcionant de la mateixa manera, seguits per un mini-joc en recompensa la TV ofereix una pel·lícula seguida d'un mini-joc
		jardí_a / coloriage_a
	joc 1 : jardí	jardí_b / coloriage_b
		jardí_c / coloriage_c
		cuina_a / memory_a
	joc 2 : cuina	cuina_b / memory_b
		cuina_c / memory_c
		dormitori_a / reparation_a
	joc 3 : dormitori	dormitori_b / reparation_b
		dormitori_c / reparation_c
	joc 4 : tv	tv / memoballons

3.3.3 Estructura final

Heus aquí l'estructura final, que serveix de base per a la versió Android, després de modificacions d'acord amb el consultor i l'autora:

TAULA 3.3: Estructura final

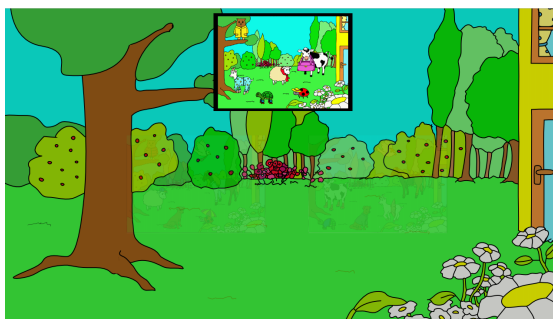
1ra escena	2na escena	3ra escena
	Decoració : hall d'entrada	
	Tria dels jocs	Menú intermediari
Animació : els 2 nens corren cap a la mare que està davant de la casa i que els invita a entrar per a jugar dins de la casa	4 zones clicables per accedir als jocs : porta del jardí porta de la cuina porta del dormitori pantalla de la TV	Cada lloc/habitació ofereix tres jocs diferents funcionant de la mateixa manera, seguits per un mini-joc en recompensa la TV ofereix una pel·lícula seguida d'un mini-joc
		jardí_a / unlock_b
	joc 1 : jardí	jardí_b / unlock_c
		jardí_c / unlock_coloriage
		cuina_a / unlock_b
	joc 2 : cuina	cuina_b / unlock_c
		cuina_c / unlock_memory
		dormitori_a / unlock_b
	joc 3 : dormitori	dormitori_b / unlock_c
		dormitori_c / unlock_répa
	joc 4 : tv	tv / memoballons

Amb aquesta última estructura, afegim l'idea de progressió al joc, en el sentit que es desbloquen noves escenes a mesura que acabem amb èxit les prèvies.

Això s'ha implementat amb l'API Web *storage* de HTML5 [3], utilitzant emmagatzemament local per a que l'informació persisteixi sempre que no es desinstal·li l'aplicació o s'en esborri les dades (és resistent als *updates*).

Nota: en l'estat, el joc té només una de cada serie d'escenes a/b/c per a cada tipus d'habitació/lloc més un de cada mini-joc (no té la variant per a la cuina del memory, mentre que sí té els balons).

FIGURA 3.4: *screenshot*: les escenes bloquejades són transparents, les altres, no



Així, per ara, l'estructura del joc queda com segueix:

TAULA 3.4: Estructura actual

1ra escena	2na escena	3ra escena
	Decoració : hall d'entrada	
	Tria dels jocs	Menú intermediari
Animació : els 2 nens corren cap a la mare que està davant de la casa i que els invita a entrar per a jugar dins de la casa	4 zones clicables per accedir als jocs : porta del jardí porta de la cuina porta del dormitori pantalla de la TV	Cada lloc/habitació ofereix tres jocs diferents funcionant de la mateixa manera, seguits per un mini-joc en recompensa la TV ofereix una pel·lícula seguida d'un mini-joc
		jardí_a / unlock_b coloriage / go_home
	joc 1 : jardí	
	joc 2 : cuina	cuina_a
		dormitori_a / unlock+go_répa répa / go_home
	joc 3 : dormitori	
	joc 4 : tv	tv / memoballons

3.4 Portatge d'aplicacions iOS cap a Android

Introduïrem aquí les diverses possibilitats que es té per a portar una aplicació iOS cap a Android, amb i sense HTML5.

3.4.1 Solucions sense HTML5

3.4.1.1 Solucions automàtiques: Apportable

Apportable és una eina “màgica” que permet convertir un espectre molt llarg d'aplicacions iOS a un apk, inclòs aplicacions utilitzant OpenGL ES, això de manera totalment automatitzada!

No obstant, per a funcions avançades tal com la projecció d'un vídeo en el context OpenGL (que s'utilitza a Tim & Timéa FR), encara no està capaç de fer-ne la conversió i genera l'error següent:

```
SpriteVideo.h:74:5: error: unknown type name 'AVAssetReaderTrackOutput'  
AVAssetReaderTrackOutput* Output;
```

3.4.1.2 Solucions manuals

La solució manual, ja discutida breument més amunt, consisteix en re-utilitzar només la part gràfica del codi, o sigui la seva part OpenGL ES, que està escrita amb el llenguatge C++ i es podria doncs incloure a una aplicació Android mitjançant el NDK, que permet fer us de codi C/C++ dins d'aplicacions Android; el contenidor, però, sempre ha de ser en Java i aquest proveiria totes les funcions àudio i de lectura de vídeo, o bé en el context OpenGL, o bé memoritzant-ho, obrint el vídeo amb l'aplicació per defecte i restituint després el context OpenGL memoritzat, com veurem que fa el *plug-in* vídeo per a Cordova (que en el seu cas memoritza/restitua el context WebView, no OpenGL, per cert).

3.4.2 Solucions utilitzant HTML5

3.4.2.1 Solucions automàtiques: Intel XDK

Una mica com Apportable, la nova versió del XDK d'Intel proporciona funcions de conversió automàtica d'aplicacions per a iOS, però aquí la destinació és HTML5 i el nombre de funcionalitats suportades és ben menor; no hi ha cap suport per a OpenGL ES.

3.4.2.2 Solucions manuals

Es la que s'emprarà, amb un bemoll: després de recrear l'aplicació des de zero amb Construct 2 utilitzant només els seus recursos multimèdia, s'utilitzarà Cordova per a contornar les limitacions encara ben presents a la implementació per Google de HTML5 al navegador Chrome *mobile* per a Android, que ve utilitzat per la classe WebView que al seu torn serveix de base per al contenidor.

3.4.2.3 Limitacions de HTML5 - solucions semi-natives

Doncs, per a contornar limitacions en quant a les capacitats de reproducció àudio/vídeo al Chrome *mobile*, utilitzarem Cordova 2.9.1 amb els *plug-ins* VideoPlayer i LowLatencyAudio, que no s'instal·la però per defecte: s'ha de prendre la versió per a Cordova 3 i fer-ne un *backport* cap a 2.9.1, ho que consisteix en modificar les declaracions d'inclusió a l'inici dels fitxers Java del *plug-in* per a que reflecteixin l'URI¹ del *package* a aquesta versió: `org.apache.cordova.api.*` (mentre que a versions 3.x, no hi ha 'api' al URI).

¹Veure la pàgina [viii](#) per una llista de les abreviacions emprades i el significat associat.

3.5 HTML5 al mòbil: estàndard(s) vs implementació

3.5.1 L'objecte àudio

Tot i que està suportat oficialment des de versions 2.x d'Android (però només amb el format mp3*), no és abans de les versions 4.x (ICS i més recent) que el *tag audio* corresponent amb l'objecte epònim va començar a suportar el format lliure ogg vorbis i ser correctament implementat per a poder utilitzar-ho en aplicacions multimèdia interactives. Tot i així, el seu suport continua essent limitat en el que fins ara, Android no va utilitzar el processador digital de só del mòbil, sinó l'unitat central de procés, per a tasques àudio! Això està canviant amb nous models de mòbils i el SO Android 4.4, ho que deixa esperar que la situació es millori a curt termini. Mentrestant, però, per a tenir un suport àudio més o menys correcte (passant pel CPU), s'ha de passar per Cordova, sobretot quan es tracta de bucles àudio, que simplement no funcionen amb HTML5 a Android. Per a aquestes, s'utilitza doncs el *plug-in* Cordova LowLatencyAudio que permet l'ús d'àudio polifònic, entre altres coses necessàries per a tenir un bucle àudio en rerefons.

S'ha modificat per a poder canviar el nivell sonor de manera dinàmica durant el joc, per exemple fer el bucle menys fort mentre que hi ha el vídeo d'introducció (només funciona a Android 4.4 - veure el comentari a la figura 3.1).

*que demana el pagament de *royalties* en certs casos, entre els quals els videojocs: veure <http://mp3licensing.com/royalty/> per a més informació.

3.5.2 L'objecte vídeo

L'objecte vídeo, molt estranyament, dintre d'una WebView, funciona quan el vídeo està a un servidor remot, però no quan està al maquinari on s'intenta obrir. A més a més, per defecte l'autoplay s'ha desactivat als navegadors mòbils, ho que posa un problema suplementari: si es vol afegir un vídeo d'introducció a l'aplicació, haurà de dependre d'una acció de touch i/o d'una font externa. Això és massa restrictiu per a Tim & Timéa, on tota l'aplicació sol funcionar sense necessitar una connexió cap a l'exterior, i la vídeo d'introducció ha de començar sense necessitar cap intervenció per part de l'usuari (en canvi, idealment ho ha de poder fer acabar més ràpidament si/quan ho vol).

Per aquests motius, s'utilitza el *plug-in* VideoPlayer, que permet assolir tot ho que s'acaba de descriure, inclosa la possibilitat per a l'usuari d'interrompre o d'acabar la visualització del vídeo si/quan ho vol.

FIGURA 3.5: *Screenshot* del vídeo pausat per l'usuari; es pot veure que s'utilitza *letterbox* per al vídeo, que no està al mateix *aspect ratio* que el joc - aquest procediment així com altres consideracions sobre els formats de pantalla es discuteixen en seguida



3.5.3 Formats de pantalla

Això és un problema que fins a l'iPhone 5 era específic a Android a causa de la gran fragmentació que hi ha al mercat dels dispositius Android, que virtualment poden tenir qualsevol *aspect-ratio*; per a evitar tenir línies blanques als confins de la pantalla, s'utilitza l'opció *letterbox* al Construct 2, ho que permet sempre tenir un *display correcte* amb bandes negres com als films 16/9 sobre televisions 4/3 quan l'*aspect-ratio* de la pantalla no està el mateix que el que hi ha per a l'aplicació, en l'ocurrència 16/9. Tanmateix, a partir de versions 4.x del SO Android a les tauletes, les barres de notificació, d'estatus i de botons *software* mai desapareixen totalment; això es pot però gestionar per a que sigui el menys problemàtic possible amb l'opció `WindowManager.LayoutParams`, que s'ha de modificar quan s'arrenca l'aplicació. Existeix també la solució SureLock que permet, en el cas de tenir accés root (super-administrador) només, passar en mode 100% *fullscreen*.

Segons la web oficial Android per a desenvolupadors, aquesta problemàtica s'ha pres en compte per a KitKat (4.4), on hi ha de nou un mode *fullscreen* integrat al SO, ho que s'ha pogut verificar amb el maquinari que es tenia a disposició (veure figura 3.4).

3.5.4 Implementació dels dibuixos animats

3.5.4.1 En general

Per a implementar dibuixos animats, està recomanat procedir amb l'anomenada tècnica *squelletal*, que consisteix en crear un esquelet per a la forma que volem animar, posar-hi textures i després només animar uns punts del esquelet per a donar l'il·lusió del moviment de tota la forma amb les textures. Aquí però, ja es disposava d'animacions Flash fetes com antics dibuixos animats, és a dir trama per trama, repetint cada vegada tots els píxels. S'ha doncs optat per re-utilitzar-les, tot i optimitzant la zona on estan actives, conservant a l'esperit el fet que també els píxels transparents ocupen memòria física quan els *sprites* s'hi descompressen per a ser visualitzats i de fet, últimament ocupen tant memòria com qualsevol altre píxel (veure figura 3.2 - tot i tractant-se d'una estimació es pot veure que la memòria utilitzada és molt més gran que la mida de l'aplicació, d'uns 45 Mo). Per a assolir-ho, s'han exportat les trames de les animacions Flash a un seguit d'imatges PNG¹, seleccionant només la zona activa. Això ha estat possible gràcies a l'ajuda de la grafista que les havia creat, atès que tenia els fitxers primaris FLA¹, sense els quals hauria estat molt més difícil fer aquesta extracció.

3.5.4.2 Al Construct 2

Al Construct 2, aquestes imatges PNG es poden i es van importar com a seguit de *sprites* formant una animació, que a l'export es transformen en un *Sprite sheet* definit al fitxer js generat per Construct 2, ho que proporciona encara una optimització respecte a l'ús de la memòria física.

A l'aplicació per a iOS, s'utilitzava una tècnica similar, tot i que els *sprite sheets* s'havien de preparar a la mà, amb un programari extern com Zwoptex per a Mac OS X.

Per a més informació sobre el subjecte, veure també <https://www.scirra.com/blog/87/under-the-hood-spritesheets-in-construct-2> i <http://software.intel.com/en-us/articles/graphics-acceleration-for-html5-and-java-script-engine-jit-optimization-for-mobile-devices>.

¹Veure la pàgina [viii](#) per una llista de les abreviacions emprades i el significat associat.

FIGURA 3.6: Exemple - fitxers generats per Construct 2 per a l'animació de la cuina



3.6 Generació vocal per mitjà de programari al entorn mòbil

Un altre canvi que s'ha introduït respecte a la versió iPhone de Tim & Timéa FR és l'ús del sistema de generació vocal pico/svox integrat a Android, que permet afegir veus allà on originalment no es van preveure, o encara canviar les veus i fins i tot l'idioma, això sense deure passar per la gravació de veu a un estudi, procés costós tant en temps que en diners. El *tradeoff* és que la veu sona una mica menys natural, però això es podria millorar comprant veus HD, sempre que estiguin compatibles amb Android.

A continuació, veurem els dos tipus principals de generadors vocals que hem considerat, i perquè s'ha triat finalment el sistema integrat a Android.

3.6.1 Generació preparada

Aquest tipus de generador és independent de la plataforma on s'executa l'aplicació que utilitza les veus, atès que equival a una gravació en estudi, amb però suposadament menys costos, tant en temps com en diners. El cost exacte depèn òbviament de si es tria un sistema gratuït, com Mbrola, que està ben adaptat per al francès, o de pago, com

el Ghostreader/iVox, que està adaptat per al francès entre altres i té unes veus HD de molt bona qualitat, incloent veus de nens, que podrien estar molt interessants en el cas de l'aplicació Tim & Timéa, i d'altres factors com la facilitat de crear fitxers àudio a partir de textos bruts, sovint inversament proporcional al preu del producte.

No obstant, no s'ha triat aquest sistema, per una banda perquè la generació preparada és menys flexible que la generació a l'execució i per l'altra, perquè l'eina pico/svox de generació a l'execució té una qualitat just entre Mbrola (relativament baixa qualitat) i les veus iVox HD per a Ghostreader, ho que es va considerar acceptable, almenys per a aquesta fase acadèmica del projecte.

3.6.2 Generació a l'execució

Al sistema Android, l'opció natural per a la generació de veu és pico/svox, que hi està integrat i s'empra per exemple per a les aplicacions de navegació de Google. Existeixen però algunes alternatives que mereixen menció, d'entre les quals l'API¹ web del acapela group es destaca particularment, pel fet de permetre la mateixa qualitat que iVox HD (que també fa part del acapela group) mitjançant una API web. A banda del fet que s'ha de pagar per al servei, la principal raó per a la qual no hem triat aquest sistema és perquè no funciona fora de línia, mentre que Tim & Timéa sol funcionar sense necessitar cap connexió exterior a ningun moment.

¹Veure la pàgina [viii](#) per una llista de les abreviacions emprades i el significat associat.

3.6.3 Implementació en complement al àudio HTML5

L'implementació que es va decidir per ara (tenint present que el producte obtingut no és un producte finit) és de completar les veus existents amb veus generades allà on la necessitat es feia sentir, és a dir sobretot als mini-jocs, que originalment tenien poc o cap instrucció vocal. Això dona lloc a uns efectes sorprenents per la barreja de veus humanes i de maquinari, de menor qualitat - ho que és el més sensible al memory amb els balons - i que no deuria quedar així a un producte comercial. Conservant l'implementació actual, la solució més apropiada sembla ser la d'utilitzar veus HD comercials compatibles amb Android, com per exemple IVONA*, idealment negociant directament amb l'empresa

que les ven un acord per a incloure-les a l'aplicació per tal d'evitar que sigui a l'usuari de comprar-les per separat.

*S'ha comprovat que amb aquesta veu HD en francès, l'aplicació funciona i la qualitat esdevé molt millor; per motius contractuals, no es pot publicar una captura àudio. No obstant, aquesta veu es pot provar gratuïtament al Google Play, buscant l'aplicació IVONA Text-to-Speech HQ i instal·lant aquesta, així com la seva veu francesa *Céline*. Instruccions per a canviar el sistema tts per defecte a Android 4+ figuren als annexos.

Capítol 4

Proves i experiments

4.1 Proves

Per tal de verificar que l'aplicació funciona com previst, s'ha procedit primer a proves individuals, al nivell de cada escena, seguides de proves globals que verifiquen el funcionament correcte del encadenament de les escenes entre elles i amb la vídeo d'introducció.

4.1.1 Proves funcionals individuals

Nota: tots els *event sheets* mencionats estan als annexos com a screenshots de Construct2

-Al jardí (escena principal):

1. Verificar que les veus funcionen correctament en cada situació, respecte al event sheet J1
2. Verificar que no hi ha problemes amb les veus quan s'afegeix la bucle àudio amb els ocells
3. Verificar que interaccions no esperades de l'usuari no provoquen comportament no desitjat
4. Verificar que l'ordre en què és diuen els noms dels vestits dels animals és aleatori (per al 1^r)

5. Verificar que l'animació de la mare amb els nens que ens feliciten funciona correctament
6. Verificar que aquesta està sincronitzada de manera acceptable amb la veu que ens felicita

-Al jardí (mini-joc):

1. Verificar que la generació vocal funciona correctament, respecte al event sheet color
2. Ídem que per al l'escena principal.
3. Ídem que per al l'escena principal.
4. Verificar que els efectes sonors (SFX) es produeixen correctament, respecte al event sheet

-A la cuina (escena principal*):

1. Verificar que les veus funcionen correctament en cada situació, respecte al event sheet Cul
2. Verificar que interaccions no esperades de l'usuari no provoquen comportament no desitjat
3. Verificar que l'animació del gat i del ratolí funciona correctament a diversos dispositius
4. Verificar que continua funcionant correctament quan s'afegeix el bucle sonor corresponent
5. Verificar que estan sincronitzats de manera acceptable: l'animació ha d'acabar-se amb el só
6. Verificar que l'animació apareix de forma aleatòria després de qualsevol objecte tret l'últim

*El memory corresponent a la cuina encara no s'ha implementat; no obstant, les proves funcionals són les mateixes per al memory amb els balons que sí s'ha implementat i es descriuran més enllà.

-Al dormitori (escena principal):

1. Verificar que les veus funcionen correctament en cada situació, respecte al event sheet Ch1
2. Verificar que interaccions no esperades de l'usuari no provoquen comportament no desitjat
3. Verificar que l'animació del conill funciona correctament, amb una temporització acceptable
4. Verificar que la música que l'acompanya s'acaba quan ja està al llit (no ha d'estar assentat)
5. Verificar que l'animació de la mare amb els nens que ens feliciten funciona correctament
6. Verificar que aquesta està sincronitzada de manera acceptable amb la veu que ens felicita
7. Verificar que l'efecte de fade al final de l'escena s'efectua correctament

-Al dormitori (mini-joc):

1. Verificar que la generació vocal funciona en acord amb el event sheet reparajoguines
2. Verificar que els efectes sonors (SFX) es produeixen correctament, respecte al event sheet
3. Verificar que l'animació final de la cortina funciona correctament amb la disminució de llum

-Al memory (mini-joc després del vídeo *L'anniversaire de Kelly*):

1. Verificar que la generació vocal funciona en acord amb el event sheet memoballons
2. Verificar que no hi hagi solapament entre la generació vocal i les veus

4.1.2 Proves funcionals globals

Aquí es verifica l'encadenament de les escenes entre elles i amb l'introducció; més específicament, els punts següents es verifiquen:

1. Després de fer totes les escenes un parell de vegades, verificar que no hi hagi incoherència
2. Fer el mateix test a almenys 3 dispositius diferents; hem emprat: Galaxy S3, Tab II i Nexus 5
3. Durant la vídeo d'introducció, verificar que el volum de la bucle àudio principal baixa

4.2 Experiments

Com anunciat a l'inici del document, la prova ultima seria de fer un *backport* de l'aplicació cap a iOS; s'ha fet, però de manera limitada, és a dir sense vídeo d'introducció ni tts. Bàsicament funciona, però falten adaptacions al nivell del *aspect-ratio*, entre altres coses. En canvi, és interessant notar que aquí el bucle àudio funciona sense necessitar un *plugin* Cordova, ho que sembla confirmar la importància de l'ús del processador digital de só del mòbil per a que això funcioni, com ho fa Apple per als seus dispositius mòbils equipats amb iOS (al contrari d'Android fins ara).

A continuació es mostren algunes captures de pantalla del que s'ha fet amb Xcode i el simulador iOS:

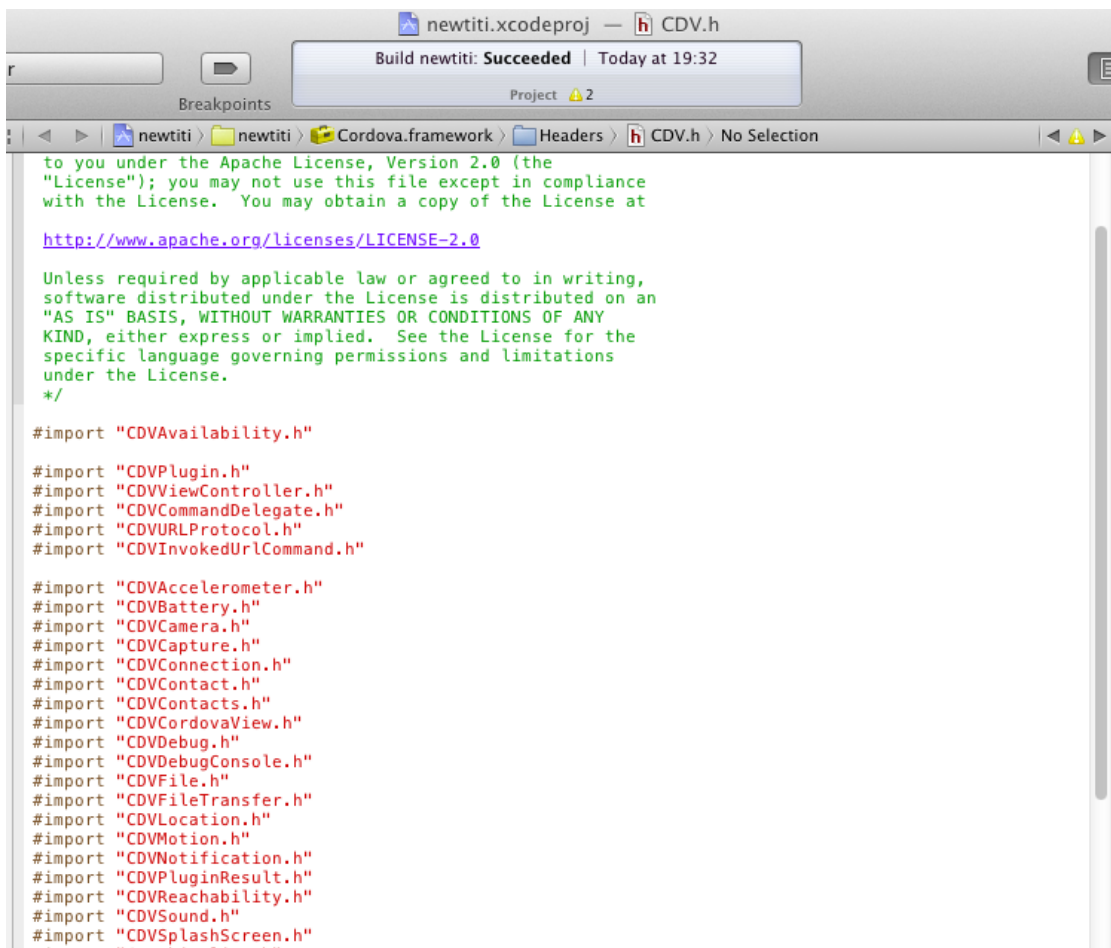
FIGURA 4.1: Xcode - build **succeeded!**

FIGURA 4.2: Entrada



FIGURA 4.3: Menú intermediari de la cuina



FIGURA 4.4: Cuina, animació del gat i del ratolí



Capítol 5

Conclusions i línies de futur

5.1 Conclusions

Podem concloure que tot i que ha encara una gran marge de progressió per al suport de HTML5 als dispositius mòbils Android actuals, amb eines complementaries adequades tal com Cordova (ex-Phonegap) es pot arribar a resultats comparables al que es pot fer amb estàndards ben establits com OpenGL ES en el domini d'aplicacions multimèdia interactives en 2D per a *e-learning*.

5.2 Línies de futur

5.2.1 Per al m-learning utilitzant HTML5

-Suport als navegadors mòbils per defecte dels SO mòbils majors d'aplicacions multimèdia interactives 100% HTML5(.1) incloent vídeo, generació de veu, etc.

-Quan es considera útil per al procés d'aprenentatge, ús de WebGL per a crear entorns tridimensionals.

-Projecció de vídeos a aquests entorns.

5.2.2 Concernint l'aplicació desenvolupada

- Una vegada la versió francesa publicada al Google Play, utilitzar la generació de veu per a d'altres idiomes.
- Portar-la a d'altres SO mòbils utilitzant la base Cordova/HTML5 desenvolupada durant aquest TFC
- Afegir contingut: noves escenes, objectes, etc.
- Afegir funcionalitats: trencaclosques, nivells de dificultat per a les funcionalitats existents
- Explorar les possibilitats d'afegir efectes gràfics més avançats amb WebGL

Bibliografija

- [1] W3C. Html5.1 editor's draft.
- [2] Dragan Peraković, Ivan Jovović, and Ivan Forenbacher. Analysis of the possibilities and effects of implementing interactive application for mobile terminal devices in m-learning system at the faculty of transport and traffic sciences. Zagreb, Croatia, 2009.
- [3] w3schools.com. Html5 web storage.