

Super Owillie. Disseny i implementació d'un joc endless running per a mòbils

Memòria de Projecte Final de Màster

Màster Universitari en Aplicacions Multimèdia

Disseny, creació i multimèdia

Autor: Elisabet Sangrà Solé

Consultor: Sergio Schvarstein Liuboschetz

17 de gener de 2017

Propietat intel·lectual

▪ Memòria, presentació i video



La memòria i presentació d'aquest projecte estan subjectes a una llicència de Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de CreativeCommons.

<https://creativecommons.org/licenses/by-nc-nd/3.0>.

▪ Joc i aplicació mòbil

© 2016 Elisabet Sangrà Solé

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

▪ Crèdits de tercers. Fonts, música i so emprats en el joc Space Owillie.

- La font **brookeshappelldots** emprada en els títols de l'aplicació mòbil ha estat creada per Brooke Shappell sota una llicència *Freeware, Non-Commercial*.
- La font **Triangulator** emprada en el cos de l'aplicació mòbil ha estat creada per Jason Pagura sota una llicència *Freeware* tant per ús personal com comercial.
- Les fonts **SF Atarian System** i **Quicksand** emprades en els caràcters numèrics de l'aplicació mòbil han estat creades per Derek Vogelpohl sota una llicència *Freeware* i Andrew Paglinawan sota una llicència *SIL Open Font License*.
- La música **Star Commander 1** emprada com a música de fons de l'aplicació mòbil ha estat produïda per DL Sounds sota una llicència *Royalty Free*.
- La música **Kick Shock** emprada com a música principal del joc ha estat produïda per Kevin MacLeod (incompetech.com) sota una llicència *Creative Commons: By Attribution 3.0 License*.
<http://creativecommons.org/licenses/by/3.0>. [Modifications: mp3 file converted into wav]
- L'efecte sonor **Balloon Popping** emprat en el joc ha estat creat per Mike König sota una llicència *Creative Commons: By Attribution 3.0 License*.
<http://creativecommons.org/licenses/by/3.0>.
- Els efectes sonors **Jingle Achievement** i **Collect Point** emprats en el joc han estat creat per LittleRobotSoundFactory sota una llicència *Creative Commons: By Attribution 3.0 License*.
<http://creativecommons.org/licenses/by/3.0>.

*Challenges are what make life interesting
and overcoming them is what makes
life meaningful.* – Joshua J. Marine

Agraïments

Aquest treball final de màster ha estat un treball desafiant, ple d'il·lusió i creativitat. Ha estat un procés complex que he aconseguit superar gràcies a totes aquelles persones que m'han acompanyat durant el camí.

Voldria agrair-vos la vostra incondicional ajuda. Sobretot a tu, Magda, sense tu tot això no hauria estat possible, una idea immaterialitzada, un no res. M'has ajudat a veure més enllà de les possibilitats en les que jo creia.

També vosaltres, companys i amics, que m'heu inspirat i heu dedicat uns moments del vostre temps a testar l'aplicació. Gràcies Gerard, Fran, Alejandro, Blai, Antonio, Tanja, Lluís i Chris.

I per últim, a tots aquells docents que m'han aconsellat i guiat durant el procés. Sobretot al consultor d'aquest treball, moltes gràcies Sergio.

Moltes gràcies a tots. Muchas gracias a todos. Vielen Dank an alle.

Abstract

Super Owillie is an app game created only with web languages.

The idea behind this project stems from the desire of developing a project which puts into practice the web development knowledge acquired during the realization of this master degree. The best way to achieve this aim is through the development of a game.

The project has been carried out in three phases: analysis, design and implementation.

The first phase is based on a brief analysis of the socio-cultural context and trends which determine today's mobile gaming market. The insights gained have been used to establish this game's scenario. The second phase designs both, the game and its wrapping app, taking into account the concept, functionalities and the entire graphic design. The third phase defines and implements the game's logic using exclusively web languages such as HTML, CSS and Javascript + P5.js library. In view of the final product, the use of hosting services and Intel XDK software are considered necessary to make the game accessible to the public.

These three complex steps led to the creation of Super Owillie, a simple game that aligns with specified functionalities but can also be enhance adding further features. In this endless running game the player controls Owillie, the main character, who travels into space. He has to react to the different dangerous obstacles that Owillie finds during his adventure by moving the owl position or changing its colour. The game's goal is to get as far as possible without crashing.

Resum

Super Owillie és una app game creada exclusivament amb llenguatges web.

La idea neix de la voluntat de desenvolupar un projecte que posi en pràctica tots els conceptes de programació web adquirits durant el màster. Es considera que la millor opció per dur a terme aquest propòsit és la creació d'un joc.

El projecte s'ha desenvolupat en tres fases: anàlisi, disseny i implementació. La fase d'anàlisi ha suposat estudiar el marc sociocultural i les tendències on es troben avui en dia els jocs mòbils. A partir d'aquí s'ha definit l'escenari on es desenvoluparà el joc. En la fase de disseny s'ha definit tant el joc com l'aplicació que l'embolcalla i se n'ha tractat el disseny gràfic d'ambdós components. Ja en la fase d'implementació, s'ha procedit a programar el joc i l'aplicació a partir d'Html, css, javascript i la llibreria P5.js. En aquesta fase i per a l'obtenció del producte final també ha estat necessari fer ús de serveis de hosting i el software Intel XDK que han permès que el joc sigui accessible per a tot el públic.

Tot plegat ha resultat un procés complex que ha donat com a resultat Super Owillie, un joc de mecànica senzilla que compleix totes les funcionalitats especificades. En aquest joc d'*endless running*, el jugador controla un mussol que viatja per l'espai. S'ha de reaccionar als diferents perills en què es troba Owillie tot movent el mussol a dreta i esquerra o bé canviant-li el color. L'objectiu del joc és arribar el més lluny possible sense col·lisionar.

Keywords / Paraules clau

App Game, Endless running, Mobile, HTML, Javascript, P5.js, Super Owillie

Notacions i Convencions

Tipografies emprades en la redacció de la memòria:

Calibri. 11 punts.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

1234567890

Normal: cos de la memòria.

Negreta: títols i paraules destacades.

Cursiva: paraules estrangeres a excepció dels noms propis.

Índex

Capítol 1: Introducció	11
1. Introducció	11
1.1 Justificació i motivació	11
2. Descripció/Definició	12
2.1 Abast	12
2.2 Lliurables	12
3. Objectius generals	13
3.1 Objectius principals	13
3.2 Objectius secundaris.....	13
4. Metodologia i procés de treball	14
5. Planificació.....	15
6. Pressupost	17
7. Estructura de la resta del document.....	18
7.1 Capítol 2: Anàlisi.....	18
7.2 Capítol 3: Disseny	18
7.3 Capítol 4: Implementació	18
7.4 Capítol 5: Demostració	18
7.5 Capítol 6: Conclusions.....	18
Capítol 2: Anàlisi	19
1. Estat de l'art	19
1.1 Jugador casual: canvi del rol del dissenyador	19
1.2 El gènere: endless running	20
1.3 L'entorn de treball.....	22
2. Especificacions i funcionalitats de l'aplicació	24
3. Llenguatges de programació i software	25
Capítol 3: Disseny	26
1. Conceptualització del joc	26
1.1 Game Design Document.....	26
2. Mecànica del joc	30
3. Arquitectura de l'aplicació	32
4. Disseny gràfic i interfícies.....	34

4.1 Interfície del joc.....	34
4.2 Interfície de l'aplicació	36
4.3 Disseny dels elements de joc.....	38
4.4 Altres: Icones, colors i fonts.....	40
Capítol 4: Implementació	42
1. Metodologia.....	42
2. Resum de la lògica del joc	44
3. Dificultats trobades	47
4. Funcions implementades	48
5. Bugs i característiques futures	54
Capítol 5: Demostració	56
1. Instruccions d'instal·lació	56
2. Mapatge de l'aplicació	58
3. Tests.....	58
3.1 Test usuari: aplicació	59
3.2 Test usuari: joc.....	59
Capítol 6: Conclusions i línies de futur.....	61
1. Conclusions	61
2. Línies de futur	63
Bibliografia	64
Annex A: Glossari	67
Annex B: Lliurables del projecte	68

Figures i taules

Índex de figures

Figura 1: App icons dels diferents jocs endless running	21
Figura 2: Captures de pantalla del joc Temple Run	22
Figura 3: Sketch de la mecànica de joc	30
Figura 4: Captura de pantalla del joc Tap Ravege– Interfície joc.....	34
Figura 5: Captures de pantalla del joc Super Owillie – Interfície joc	35
Figura 6: Captures de pantalla del joc Super Owillie - Menús	36
Figura 7: Captures de pantalla del joc Mooniz – Pantalla d’inici	37
Figura 8: Captures de pantalla del joc Super Owillie – Pantalla d’inici.....	37
Figura 8: Captures de pantalla del joc Super Owillie – Pantalles aplicació.....	38
Figura 10: Disseny dels elements de joc - Owillie.....	38
Figura 11: Disseny dels elements de joc - Meteorit.....	39
Figura 12: Disseny dels elements de joc - Planetes	39
Figura 13: Disseny dels elements de joc - Estrelles	39
Figura 14: Icona de l’aplicació Super Owillie	40
Figura 15: Exemples tipogràfics	41
Figura 16: Captura de pantalla del joc Angry Birds - Nivells	55
Figura 17: QR Super Owillie Web App	56

Índex de diagrames

Diagrama 1: Planificació Gantt	16
Diagrama 2: Mecànica del joc Super Owillie	31
Diagrama 3: Arbre de navegació de l’aplicació Super Owillie	32
Diagrama 4: Mapatge i Navegació de l’aplicació Super Owillie.....	58

Índex de taules

Taula 1: Taula de fites	15
Taula 2: Estimació pressupost projecte	17
Taula 3: Jocs més populars d’endless running	21
Taula 4: Super Owillie – Personatges i Elements	26
Taula 5: Super Owillie – Taula de nivells	28
Taula 6: Super Owillie – Puntuacions	28
Taula 7: Funcions implementades	53

Capítol 1: Introducció

1. Introducció

Els dispositius mòbils han esdevingut elements imprescindibles en l'era digital. Han tingut un gran impacte econòmic i social en la societat i han influït en diversos sectors i indústries. La indústria del videojoc ha evolucionat tant, gràcies a l'aparició d'aquests dispositius, que ara ja es parla d'una nova indústria: la *Mobile Gaming*.

Precisament aquest projecte de final de màster ha consistit en la creació d'un producte competent per a aquesta indústria. S'ha dissenyat i implementat un joc per a mòbil a partir dels llenguatges web (html5, css i javascript). L'elecció dels llenguatges responen a la intenció de posar en pràctica allò après durant el màster i a la intenció que aquest mateix codi pugui ser adaptat en diversos dispositius independentment del sistema operatiu (iOs/Android).

El joc ha protagonitzat la part més àmplia de programació, no obstant això, el projecte també ha tractat el disseny d'interfícies donant com a resultat un joc i l'aplicació que l'embolcalla.

1.1 Justificació i motivació

Aquest és un projecte que pretén satisfer la necessitat d'oci i entreteniment. El món dels videojocs ha canviat, ha canviat el tipus de jugadors i amb ells les regles del mercat. Segueixen sent els joves els que dominen el mercat però la *Mobile Gaming* també arriba a un sector adult de mitjana edat (entre 40-50 anys). Aquesta és la gran diferència amb els videojocs tradicionals, una diferència que influeix en com ha de ser el *gameplay* d'aquests jocs.

Els jocs per a dispositius mòbils són jocs de mecànica senzilla, amb unes regles molt bàsiques i intuïtives. Això és comprensible tenint en compte que la mitjana de temps que es dedica al *gaming* és d'uns 7,5 minuts per sessió. Els usuaris dediquen poc temps a jugar amb el mòbil, així que els jocs no poden ser gaire complexos ni narratius.

Aquest és el context idoni per a desenvolupar el treball final de màster, ja que una *app game* permet tant el disseny de l'aplicació com el repte de crear un joc simple a partir dels llenguatges de programació web.

2. Descripció/Definició

El màster universitari en Aplicacions Multimèdia culmina amb la presentació d'un treball final que en aquest cas abasta el disseny i implementació d'un joc *endless running* per a mòbils.

És un treball que vol crear des de zero una aplicació mòbil que funcioni en diversos dispositius independentment del sistema operatiu.

Avui en dia quan es crea una aplicació és important definir en quina plataforma s'executarà. El més probable és que es vulgui que l'aplicació funcioni tant per a plataformes iOS com Android. En aquest cas, usar una programació nativa pot suposar una pèrdua de temps que alhora es tradueix en una pèrdua econòmica i de recursos. Una solució a aquest problema és usar els llenguatges web ja que a partir d'aquests es possible generar una única aplicació que funcioni en ambdues plataformes.

D'aquesta manera, per a crear l'aplicació mòbil s'empraran els llenguatges web apresos al llarg del màster: HTML, CSS, Javascript i Processing. I s'intentarà demostrar com amb aquests llenguatges no nadius realment es pot crear una aplicació per a tots els dispositius mòbils.

L'aplicació mòbil que es desenvoluparà serà una *game app*, és a dir, un joc per a mòbil.

2.1 Abast

Es pretén dissenyar i implementar un joc a partir de llenguatges web que funcioni en mòbils. Donat el temps i els recursos dels quals es disposa, el joc haurà de tenir una mecànica senzilla per això s'ha escollit crear un joc del gènere Endless Running ja que aquest es caracteritza per uns controls limitats i amb poques accions.

2.2 Lliurables

El projecte inclourà una sèrie de lliurables:

- informes de seguiment
- memòria
- presentació acadèmica en format vídeo
- presentació pública en format power point
- una web app
- codi-font de l'aplicació i joc

3. Objectius generals

L'objectiu principal d'aquest projecte és posar en pràctica tots els coneixements apresos durant el màster en dissenyar i implementar un joc *endless running* per a mòbils.

Tot i que aquest projecte se centra en què el joc funcioni en mòbils, es farà ús de llenguatges web per tal que en un futur es pugui adaptar fàcilment a altres dispositius ja siguin tabletas o navegadors.

3.1 Objectius principals

Objectius del producte

- Implementar no tan sols el joc si no també l'aplicació que l'engloba.
- Donar a l'aplicació i joc un estil divertit i amè. Quelcom que entri per la vista.
- Crear un joc que segueixi les tendències del mercat però que alhora aporti alguna novetat en el *gameplay*.
- Adaptar el gènere per a poder incloure-li nivells.
- Crear una corba de dificultat adequada.

Objectius per a l'usuari

- Buscar un perfil de jugador casual.
- Crear un joc que sigui entretingut, additiu i que tingui ritme.

Objectius personals

- Dur els coneixements de programació apresos al territori dels jocs, món que em captiva cada cop més.

3.2 Objectius secundaris

Objectius secundaris i de futur

- Adaptar l'aplicació a altres plataformes.
- Adquirir els drets de fonts i músiques emprats en l'aplicació.
- Distribuir l'aplicació en les principals botigues: AppStore i/o Google Play (com a aplicació gratuïta).

4. Metodologia i procés de treball

Aquest és un projecte que se centra en el desenvolupament d'una aplicació. D'una banda, presenta un joc com a element principal i d'altra banda, una sèrie de pantalles i funcionalitats que envolten el joc tot originant una aplicació.

Com tot projecte de desenvolupament tècnic, aquest compta amb dues grans etapes: disseny i implementació. En l'etapa de disseny s'ha pensat de forma teòrica en com serà el joc i l'aplicació. En l'etapa d'implementació s'ha materialitzat aquesta idea.

Es tracta d'un producte que parteix des de zero tot i que en l'actualitat existeixen productes semblants. Per això hi ha hagut una etapa prèvia d'anàlisi i definició.

El procés de treball ha quedat definit de la següent manera:

1. Definició del projecte: *scope*, objectius, planificació...
2. Anàlisi de la indústria del joc mòbil a partir de la qual s'ha escollit una temàtica de joc adient a les tendències actuals.
3. Anàlisi dels entorns de treball i elecció de les eines que s'empraran en l'etapa d'implementació.
4. Conceptualització del joc i de la seva mecànica.
5. Disseny d'interfícies de l'aplicació tot definint les funcionalitats i la navegació.
6. Implementació del joc i de l'aplicació fent ús de llenguatges web.
7. Proves d'usuari tant de l'aplicació com del joc i, posteriorment, realitzar les correccions oportunes.

Una part important per validar de forma objectiva el producte ha estat la realització de les proves d'usuari. Aquestes han consistit en dues fases:

- Primer s'ha testat la navegació de l'aplicació. S'ha demanat a 10 persones que realitzessin una sèrie de tasques a partir del prototip realitzat. I se n'ha analitzat el comportament dels usuaris i si aquest era previsible.
- Segon s'ha testat el funcionament del joc i de les pantalles que actualitzen dades. S'ha demanat a 20 persones que provessin el joc i que ens donessin la seva opinió. Què els hi havia agradat i què fallava.

5. Planificació

El projecte està especialment marcat per l'entrega de les diferents PACs, cadascuna d'aquestes entregues marca una etapa i n'estableix les principals fites. Val a dir que el projecte s'ha allargat 8 dies respecte a la planificació inicial.

TAULA DE FITES			
Nom	Durada	Inici	Final
PAC 1 – Proposta	15 dies	26.09.2016	10.10.2016
PAC 2 – Mandat del projecte i planificació	14 dies	11.10.2016	24.10.2016
PAC 3 – Entrega 1 : Anàlisi i Disseny	28 dies	25.10.2016	21.11.2016
PAC 4 – Entrega 2: Implementació	30 dies	22.11.2016	21.12.2016
PAC 5 - Tancament	27 dies	22.12.2016	17.01.2017
Defensa del projecte	14 dies	10.01.2017	27.01.2017

Taula 1: Taula de fites

Pel que fa la planificació, i seguint les etapes marcades per les diferents PACs, el projecte està dividit en 6 parts diferenciades:

Proposta

Corresponent a la PAC 1. Consta del primer debat on es proposa i s'aprova el tema del TFM.

Mandat del projecte i planificació

Corresponent a la PAC 2. Tracta la definició del projecte i integra: l'*scope* del projecte, la descripció d'objectius i la planificació de tot el projecte.

Entrega 1 : Anàlisi i Disseny

Corresponent a la PAC 3 i inclou un gruix considerable de feina. Aquesta etapa representa un primer contacte amb l'aplicació i el joc que es vol desenvolupar en termes de disseny i conceptualització. També és una etapa d'anàlisi en la que s'ha d'escollir la millor eina per treballar.

Entrega 2: Implementació

Corresponent a la PAC 4 i també amb un gruix considerable de feina. Aquesta és la part més pràctica ja que consisteix en crear a partir de codi tot allò especificat en l'etapa de disseny.

Tancament

Corresponent a la PAC 5. És el tancament del projecte la qual cosa inclou finalitzar la implementació del joc, revisar-lo i corregir-lo. És important realitzar les proves d'usuari i fer els últims retocs abans d'entregar el producte.

Defensa del projecte

A continuació es pot veure amb més detall la planificació a través del diagrama de Gantt del projecte:

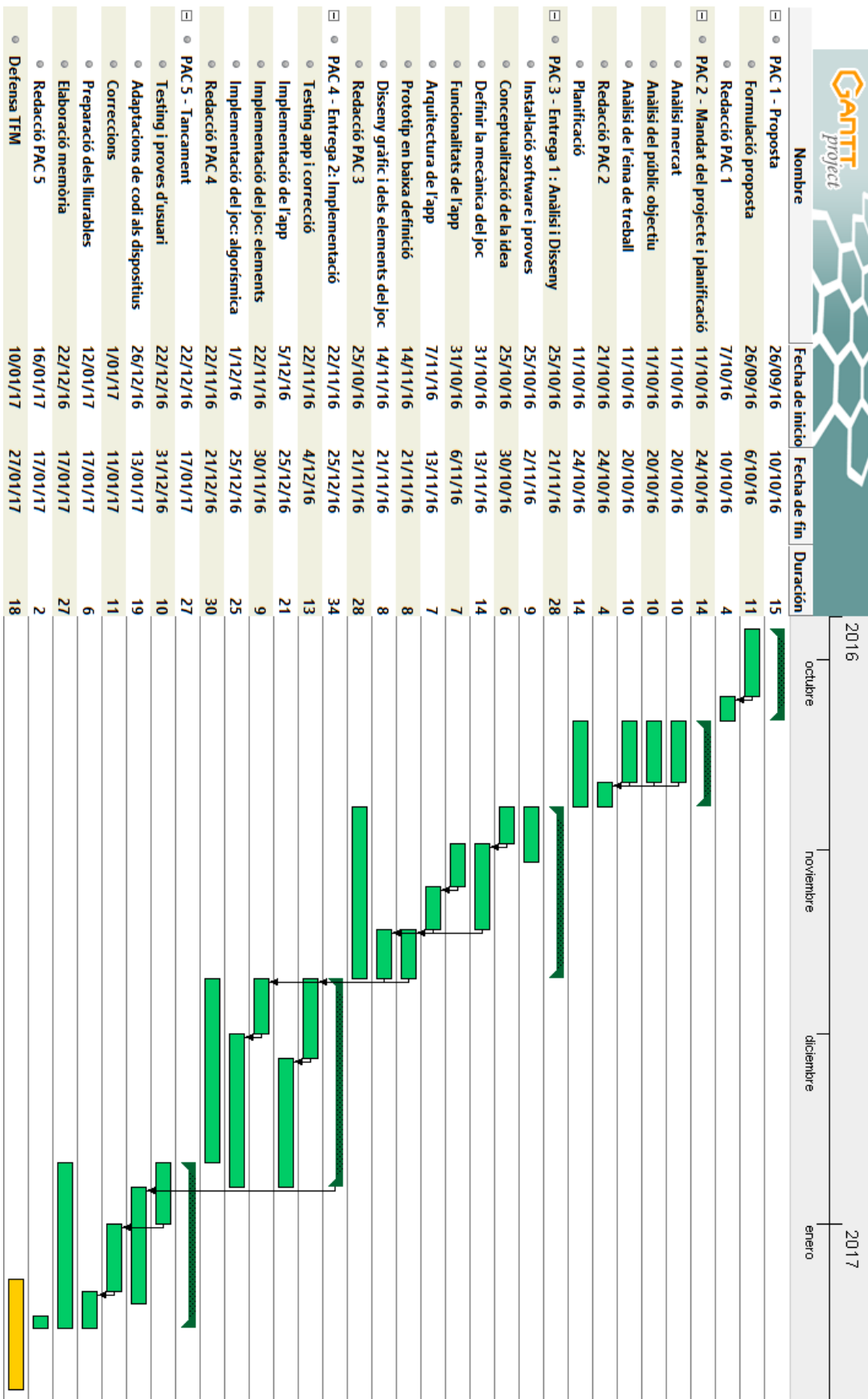


Diagrama 2: Planificació Gantt

6. Pressupost

El cost real d'aquest projecte ha estat de 170€ equivalents a la reparació de l'ordinador principal. No s'ha hagut d'adquirir cap *hardware* addicional i s'ha fet ús de llicències de *software* ja adquirides temps enrere. Pel que fa el personal, la part corresponent al dissenyador web i al programador web han estat assumides per mi mateixa. Les úniques persones que han col·laborat ho han fet en qualitat de testers i de forma gratuïta.

Donat que no hi ha hagut un pressupost fixat, a continuació es mostra una estimació del que podria haver estat el cost mínim d'un projecte com aquest.

6.1 Equip Humà

Personal	Tasca	Hores	Tarifa	Total
Dissenyador Web	Disseny de l'aplicació i dels elements del joc.	20	20€/h	400 €
Programador Web	Implementació de l'aplicació i del joc.	100	25€/h	2500 €

6.2 Equipament Tècnic i Software

Equipament	Preu
Ordinador amb pantalla tàctil - Windows 8	1800 €
External Hard Disk	59 €
Llicència Microsoft office	69 €
Llicència Dreamweaver	290 €
Balsamic Mockup	Gratuït (1 mes)
Entorn de treball: Intel XDK, Phone Gap, Android Studio...	Gratuït
Serveis de <i>hosting</i> – 000webhost.com	Gratuït

6.3 Altres despeses

Despesa extra	Hores	Preu/Hora	Total
Personal per les proves d'usuari (x15)	2	10 €	300 €
Adquisició dels drets d'ús de música, FX i fonts	-	-	200 €

6.4 Total

TOTAL	5618 €
--------------	---------------

Taula 2: Estimació pressupost projecte

7. Estructura de la resta del document

7.1 Capítol 2: Anàlisi

Aquest capítol respon a la necessitat d'estudiar el panorama del joc mòbil actual, tot definint el tipus de jugador i el gènere *endless running*. Recull també les especificacions del producte arrel d'haver analitzat jocs semblants i exposa els llenguatges i eines que s'usaran.

Es podria dir que és un capítol que serveix per contextualitzar i establir les bases per desenvolupar el projecte.

7.2 Capítol 3: Disseny

Aquest és un capítol que defineix de forma teòrica el producte que es vol crear. És el capítol més creatiu i bàsicament tracta dos aspectes fonamentals: el disseny del joc (concepte, regles, aspecte gràfic...) i el disseny de l'aplicació (pantalles, navegació, aspecte gràfic...).

7.3 Capítol 4: Implementació

Aquest capítol pretén ser un resum sobre el procés d'implementació del joc i aplicació. Explica les decisions preses i fa un recorregut sobre les funcions implementades.

7.4 Capítol 5: Demostració

Aquest capítol emmarca el producte final. D'una banda estableix una sèrie d'apartats que serveixen com guia de suport per mostrar el producte final. D'altra banda inclou comentaris del producte obtinguts durant les proves d'usuari.

7.5 Capítol 6: Conclusions

Un capítol de reflexió on es valora el que ha suposat l'elaboració d'aquest treball tant a nivell global com a nivell personal.

Capítol 2: Anàlisi

1. Estat de l'art

Els videojocs estan travessant la seva segona època daurada gràcies a l'exorbitant expansió i popularitat dels *smartphones* i de la seva tecnologia *swipe* usada en les pantalles tàctils. Els *smartphones*, a més, han suposat la gran revolució de l'entreteniment digital fins a convertir-se en un gran fenomen global que ha permès arribar a un públic més extens que les consoles tradicionals. És per això que el mercat ha canviat així com també ho ha fet el seu *target*, ens trobem ara davant d'un jugador casual.

1.1 Jugador casual: canvi del rol del dissenyador

L'aparició d'un gran nombre de jugadors casuals que estan interessats en els jocs mòbils fan replantejar les regles de disseny. Tenint en compte que el jugador casual és un tipus de jugador que no inverteix gaire temps en jocs i que només hi juga ocasionalment, els jocs no poden ser tractats de la mateixa manera en què es tracten en les consoles. Els jocs mòbils actuals s'han d'adaptar a les demandes d'aquest tipus de consumidors.

Això suposa desenvolupar un joc tipus *pick up & play*, és a dir, agafar i jugar. Es tracta de jocs amb regles simples i fàcilment recordables, doncs pot ser que el jugador casual no torni a jugar al mateix joc fins passat un llarg temps. La jugabilitat ha de ser senzilla i amb uns controls intuïtius. El *gameplay* ha de ser additiu i s'ha d'evitar introduir-hi gaires accions per no fer el joc és massa complex.

Aquests tipus de jocs han de tenir una ràpida corba d'aprenentatge perquè sinó el joc serà avorrit i augmentarà la probabilitat de fracàs. El joc ha de seduir des d'un primer moment perquè el mercat és tant ampli i hi ha tantes opcions que el jugador l'esborrarà ràpidament del mòbil si no li agrada. En aquesta mateixa línia, el joc no ha de ser ni massa fàcil ni massa difícil, un bon equilibri de nivell-dificultat garanteix que el joc sigui divertit.

Tot i que el més important en aquest tipus de jocs casual és el *gameplay*, un altre factor important és el disseny gràfic. D'aquests jocs tipus casual no s'espera tenir uns gràfics espectaculars però sí una estètica atractiva i, encara que depenent del joc, probablement amb colors vius i personatges empàtics.

Ara bé, no tot gira entorn del joc en sí. Jugar és una experiència i més enllà de la mecànica, el joc i l'aplicació poden aportar funcionalitats addicionals com pot ser l'obtenció de *power ups* o objectes especials, superació de reptes, desbloquejar nivells... que el que fan és afegir incentius per al jugador.

Es pot dir que ha sorgit una nova manera de concebre els videojocs on el que se cerca és la simplicitat combinant-la amb una bona dosi de diversió i creativitat. La indústria del videojoc s'ha expandit fins al punt que els jocs mòbils disposen d'una indústria pròpia, la *Mobile Gaming*. Dins d'aquesta nova indústria hi ha certs gèneres que han triomfat més que uns altres. Un dels més estesos avui en dia és l'*Endless Running*.

1.2 El gènere: *endless running*

L'*Endless running* -traduït de l'anglès com "córrer sense fi"- és un subgènere dels jocs de plataforma que es caracteritza perquè el personatge principal és mou constantment cap endavant. En aquest tipus de joc, els controls són força limitats i tan sols es contemplen accions senzilles com saltar, esquivar, atacar o enderrocar. L'objectiu del joc és arribar el més lluny possible abans de morir.

El gènere d'*endless running* ha estat present en la història dels videojocs des dels primers temps. *Arcade* jocs com *Wheels* (1974) ja usaven aquest estil de món infinit, un estil conegut aleshores com *side-scrolling*. També les primeres videoconsoles presentaven títols d'*endless running* com *Moonpatrol* (1982) o *Antartic adventure* (1985). Popularment i en consoles de generacions més avançades trobem el cèlebre joc de *Crash Bandicoot* (1996). Però no és fins a principis de 2003 quan apareix NOM, el primer *endless running* exclusiu per a mòbil. L'èxit i posterior influència de NOM va venir gràcies al seu senzill *gameplay*, doncs els controls estaven limitats a un únic botó. Això va permetre una millor interactivitat mòbil-persona.

Aquest gènere s'ha fet molt popular en els últims anys, sobretot perquè és un gènere que encaixa molt bé amb la filosofia mòbil. Des de 2010 s'han produït una gran quantitat de jocs *endless running* per a plataformes mòbils, alguns d'aquests jocs han tingut una gran acceptació fins al punt que han portat el gènere a la popularitat.

Alguns dels més destacats són:








						
Temple Run	Subway Surfers	Minion Rush	Jetpack Joyride	One Epic Knight	Ski Safari	Banana Kong

Figura 1: App icons dels diferents jocs endless running

Taula 3: Jocs més populars d'endless running

De tots aquests, Temple Run^[3] es mereix una menció en especial ja que és el joc per excel·lència que ha popularitzat aquest gènere en els dispositius mòbils.

Història

Guy Dangerous és un arqueòleg que busca un tresor en els jardins d'un temple. Malauradament és perseguit per una sèrie de micos endimoniats que se'l volen menjar.

Objectiu del joc

Els jardins del temple no tenen final així que el Guy Dangerous corre fins que cau per algun forat, col·lisiona amb algun arbre/roca o és capturat pels micos endimoniats. L'objectiu principal és arribar el més lluny possible i aconseguir quantes més monedes millor.

És tracta d'un joc senzill on el jugador és incentivat a tornar-hi a jugar per superar la seva pròpia puntuació, tal com passava amb les màquines d'*arcade*.

Game play

El jugador controla a l'arqueòleg Guy Dangerous que el pot moure en tres posicions (esquerra, mig, dreta) tot fent ús de l'acceleròmetre del *smartphone*. Aquest moviment serveix sobretot per recollir monedes. La pantalla tàctil l'usa amb els diferents *swipes* per: saltar, passar per sota o girar a dreta i esquerra.





Figura 2: Captures de pantalla del joc Temple Run

Temple Run és un clar exemple de com un joc amb uns gràfics atractius i amb una mecànica molt senzilla pot tenir un gran impacte en la indústria del videojoc mòbil. El gènere d'*endless running* està en auge i és perfecte per a aquest tipus de jugador casual, sobretot perquè permet jugar en curtes ràfegues de temps i perquè les accions són limitades.

1.3 L'entorn de treball

En l'actualitat coexisteixen diverses plataformes mòbils. Les més populars són iOS i Android però també n'hi ha d'altres de menys populars com Windows Phone, Blackberry, Tizen o Firefox OS. Cadascuna d'aquestes plataformes tenen un sistema operatiu diferent i empen un llenguatge específic.

Al principi això va ser tot un problema pels dissenyadors ja que havien de programar una versió de la mateixa aplicació per cadascun dels diferents sistemes, la qual cosa suposava un esforç extra i una certa pèrdua de temps. Una ràpida solució va ser utilitzar algun *framework* intermedi que funcionés en tots els sistemes i que d'aquesta manera es pogués reutilitzar la lògica de l'aplicació.

Sens dubte, una de les millors solucions avui en dia és utilitzar llenguatges web i utilitzar algun host multiplataforma. *Write once, run everywhere.*

Els objectius d'aquest projecte contempen implementar el joc a partir de llenguatges web. Precisament per aquest motiu, perquè d'aquesta manera és més senzill crear un joc que s'adapti i

funcioni en les diferents plataformes sense necessitat de conèixer el llenguatge natiu de cadascun dels sistemes operatius mòbils.

Una petita part d'aquest projecte s'ha centrat en conèixer quins són els entorns de treball actuals que permeten desenvolupar aplicacions multiplataforma per a mòbils. Dins dels més populars trobem Phone Gap, Cordova Apache, Intel XDK, Titanium Mobile, AngularJs i Ionic. Aquests ofereixen una sèrie de kits de treball que permeten de forma senzilla crear aplicacions natives per a cadascun dels diferents sistemes.

2. Especificacions i funcionalitats de l'aplicació

Avui en dia hi ha tanta competència en el mercat de les *games apps* que un simple joc no és suficient. El joc ha d'anar més enllà, i per això ha d'incloure una sèrie d'elements extres que el complementen i n'enriqueixen l'experiència.

Analitzant els jocs més populars del mercat s'observen diversos elements, a més del joc, que han estat factors clau del seu èxit.

Leaderboards - Highscore - Stats - Achievables i Objectius - Share on Social Media - Desbloqueig d'ítems - Power ups - Multiplier

Aquests elements addicionals no fan altra cosa que incrementar el nivell de repte i oferir una ajuda per superar-lo.

Donat que en aquest treball de fi de màster s'ha realitzat una primera versió del joc, s'han exclòs elements socials, *power ups* i desbloqueig d'ítems. Ara bé, sí que s'ha intentat motivar el jugador a través de les puntuacions i així fer augmentar el nivell de repte. Això no és una novetat, ja ve dels jocs d'*arcade* on un dels factors que feia aquests jocs tant additius era la idea de "superar la millor puntuació" i aparèixer en el *top 10*.

A continuació es defineixen les funcionalitats de l'app a implementar:

- Dos modes de joc: un mode clàssic d'*endless runner* i un mode amb nivells.
- Els nivells estan bloquejats i es desbloquegen a mesura que s'avança en el joc.
- El joc presenta elements que fan augmentar la puntuació.
- El joc mostra part de la puntuació durant la partida i al finalitzar el joc se li sumen els extres.
- Es poden consultar les millors puntuacions.
- La música i efectes sonors poden ser desactivats.

Al tractar-se d'una primera versió i considerant el temps del qual s'ha disposat, el projecte s'ha centrat en el cos del joc i els elements bàsics. Així que s'han descartat tota una sèrie d'elements i funcionalitats:

- Elements com *Achievables, Power ups, multipliers...*
- Component social
- No s'implementa una base de dades, i per tant no té sentit tenir un *Leaderboard*.
- Tampoc té sentit una pantalla per a comprar ítems o de descàrrega d'altres jocs (com moltes aplicacions tenen).
- *Sign up* o *Log in*.

3. Llenguatges de programació i software

Una de les motivacions principals per haver escollit realitzar una *game app* com a projecte final és l'interès personal en l'àmbit de la programació. Des de l'inici del projecte ja s'ha definit que la part tècnica d'aquest es resoluria a partir de llenguatges web treballats durant el màster.

Llenguatges web

- **HTML** vist en l'assignatura "Llenguatges i estàndards web" (complement de formació).
- **CSS** vist en l'assignatura "Llenguatges i estàndards web" (complement de formació).
- **Javascript** vist en l'assignatura "Tecnologies i Eines pel desenvolupament web" i en "Programació web" (complement de formació).
- **P5.js** vist en l'assignatura "Programació creativa"

Per a la resolució tècnica de l'aplicació s'han emprat diversos llenguatges web: html, css, javascript però sobretot la biblioteca de processing, P5.js. S'ha escollit aquesta biblioteca per a desenvolupar el joc perquè ofereix la possibilitat d'usar el canvas i perquè, a diferència de javascript, presenta funcions bàsiques de dibuix més senzilles d'usar com ellipse() per crear un cercle o rect() per crear un quadrat. D'aquesta manera la part gràfica resulta menys laboriosa que usant exclusivament javascript.

Software

- **Balsamiq mockups**: *wireframing software* destinat a la creació de prototips en baixa definició.
- **Dreamweaver**: entorn per a escriure el codi.
- **Intel XDK o Android Studio**: entorn de treball per a empaquetar l'aplicació.

Tot i que en primer lloc es va plantejar treballar exclusivament amb l'entorn d'Intel, finalment es va considerar treballar amb Adobe Dreamwaver per escriure el codi. El motiu és la prèvia familiarització amb aquesta eina. De totes maneres al final no s'ha usat l'entorn d'Intel ni cap entorn semblant.

Serveis de hosting

- **000webhost**: servei gratuït de *hosting* on s'han pujat els arxius de l'aplicació per accedir-hi des d'un navegador. Aquesta eina a més de gratuïta és senzilla d'usar ja que utilitza una interfície més de web 2.0 amb la qual cosa no requereix d'un servei addicional de FTP.
<https://es.000webhost.com/>

Capítol 3: Disseny

1. Conceptualització del joc

El primer pas és donar forma a la idea inicial, és a dir, definir tots els detalls del joc per així poder implementar-los en la següent fase. El punt de partida és el *Game Design Document*, un document que tracta diversos aspectes del joc des d'una perspectiva purament descriptiva. Aquest document servirà de guia per al desenvolupament del joc.

1.1 *Game Design Document*

Descripció inicial

Space Owillie és un joc *endless running* per a mòbil.

Història

Owillie és l'últim mussol que queda a la Terra. Els seus amics ja fa temps que van creuar la nebulosa per anar a viure a un altre planeta. Avorrit cada nit de contemplar tot sol la lluna, decideix engrescar-se en una aventura còsmica en busca d'aquest nou planeta i dels seus amics. El que no sap és que l'univers no és tant segur com sembla i es trobarà amb diversos perills que haurà d'evitar.

Personatges i Elements

Owillie	Personatge principal. Pot canviar de color. És controlat pel jugador.
Meteor	Elements de tamany reduït. S'han d'esquivar.
Planets	Elements grans i de diferents colors. Poden ser esquivats o bé travessats si Owillie presenta el mateix color que el planeta.
Stars	Elements per recol·lectar. Donen punts addicionals.
Friendly Owls	Elements per recol·lectar (3 per nivell). Donen punts addicionals i desbloquegen nivells.

Taula 4: Super Owillie – Personatges i Elements

Gameplay

El jugador pren el control d'un mussol anomenat Owillie que viatja per l'espai direcció a un nou planeta on s'hi han instal·lat tots els seus amics. Durant el viatge, aniran apareixent en la pantalla una sèrie d'elements que s'han d'esquivar o en cas contrari Owillie s'hi estavellarà i perdrà el rumb.

L'objectiu és mantenir el rumb correcte durant el màxim temps possible. Per a aconseguir-ho, el jugador pot moure Owillie a dreta i esquerra per esquivar els objectes o, si es tracta d'un planeta, li pot canviar el color per travessar-los. La partida acaba quan Owillie impacta amb aquests objectes i, per tant, perd el rumb (el que equivaldria a què el personatge mor).

Durant la partida també apareixen altres elements que en comptes d'evitar s'han de recol·lectar per fer augmentar la puntuació.

Així doncs, hi ha tres accions que el jugador pot realitzar:

- Esquivar meteorits i planetes tot desplaçant el mussol a dreta o esquerra en aquelles zones on hi hagi buit.
- Travessar planetes només en el cas que el mussol i el planeta siguin del mateix color. Per això el jugador pot en tot moment canviar el color del mussol.
- Recol·lectar estrelles i mussols amics impactant amb ells, és a dir, en aquest cas no s'han d'esquivar sinó intentar moure el mussol allí on hi ha un d'aquests dos objectes.

Game Controls

El joc busca un perfil de jugador més casual amb la qual cosa els controls no poden ser complicats ja que aquest és un tipus de jugador que vol obrir l'aplicació i seguidament jugar-hi. No hi volen dedicar temps a aprendre com s'hi juga.

Aquest és el motiu pel qual el joc només tindrà dos moviments:

- *Swipe right/left*: per controlar el moviment del mussol.
- *Tap*: per canviar el color del mussol i per seleccionar/acceptar les diverses opcions.

Modes de joc

El joc inclourà dos modes:

- *Play Classic*: un mode sense final on la partida s'acabarà quan el mussol impacti amb un meteorit/planeta. En aquest cas la dificultat vindrà donada per un augment en la velocitat en la que els elements es mouen per la pantalla.

- *Play Levels*: un mode amb nivells on hi haurà un final de partida. Cada nivell inclourà una dificultat addicional en forma d'augment de velocitat o afegir nous colors.

En aquest darrer mode hi haurà 27 nivells, els tres primers estaran desbloquejats i els restants s'hauran de desbloquejar. Per desbloquejar els nivells s'haurà d'aconseguir un cert nombre de mussols amics (n'hi ha tres per nivell).

<i>Levels</i>		
Level 1	<i>Unlocked</i>	-
Level 2	<i>Unlocked</i>	-
Level 3	<i>Unlocked</i>	-
Level 4	<i>Locked</i>	<i>6 owls to unlock</i>
Level 5	<i>Locked</i>	<i>8 owls to unlock</i>
Level 6	<i>Locked</i>	<i>10 owls to unlock</i>
...
Level n	<i>Locked</i>	<i>((n*2)-2) owls to unlock</i>

Taula 5: Super Owillie – Taula de nivells

El recompte de mussols amics només es realitzarà si s'ha completat el nivell. En cas d'haver aconseguit un mussol però no haver completat el nivell, aquest mussol no servirà per desbloquejar el següent nivell.

Puntuació

En un joc d'*endless running* és important que cada segon de joc compti i faci incrementar la puntuació. Es considerarà aquesta puntuació com els metres recorreguts. Addicionalment s'hi afegiran extres com punts per estrella recol·lectada o per mussol amic recollit.

Els metres recorreguts es mostraran durant la partida però els extres s'hi afegiran al final de la partida que es mostraran en una altra pantalla. El sistema de puntuatge definit és el següent:

<i>Score</i>	
Metres recorreguts (<i>every draw() interval</i>)	+1
Stars 4 points	x100
Stars 5 points	x300
Friendly Owl	x500

Taula 6: Super Owillie – Puntuacions

Art

Algunes pinzellades sobre el tema artístic són:

- El joc s'ambientarà i presentarà elements de l'univers ja que l'acció transcorre a l'espai exterior.
- A nivell estètic es pretén allunyar d'una estètica retro i/o d'*arcade* i crear quelcom divertit, amè i amb un to desenfadat. Un exemple a tenir en compte és la part estètica d'Angry Birds que aconsegueix connectar amb el jugador a través de la caracterització dels seus personatges.
- Sobre els colors aquests seran vius en contrast amb un fons fosc.
- S'haurà de crear la sensació de moviment a partir de les velocitats en què es mouen els objectes i, intentar animar el fons a partir d'elements mòbils.

Música i sons

El joc comptarà amb música de fons i efectes sonors quan el mussol s'estavelli, reculli un objecte o completi un nivell.

A l'hora d'obtenir les diferents peces musicals s'intentarà buscar *hits* lliure de drets o que se'n permeti fer un ús gratuït. Llibreries online com <http://www.dl-sounds.com>, <http://www.freeplaymusic.com> i <https://www.freesound.org> ofereixen músiques amb *royalty free*.

Referències

- Alguns jocs destacats del gènere endless running: Temple Run, Super Mario Run (data de llançament: desembre), Banana Kong, One Epic Knight, Sky Jumper, Bird Jumper, Sky on Neon.
- Alguns jocs que han marcat el mercat del Mobile Gaming: Angry Birds, Candy Crush.

2. Mecànica del joc

Aquest apartat pretén definir les regles del joc. Donat que es tracta d'una idea de joc senzilla, la mecànica del joc ha quedat força explicada en el *gameplay*.

De totes maneres el més important a considerar és:

- El jugador controla el personatge principal que és un mussol anomenat Owillie.
- El jugador pot moure el mussol a dreta i esquerra tot lliscant el dit per la pantalla en la direcció desitjada. S'ha de tenir en compte que el personatge només es pot moure horitzontalment en tres posicions.

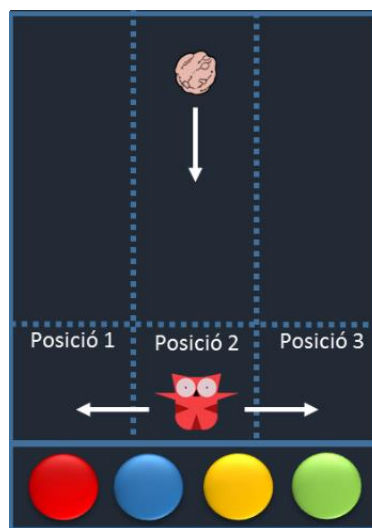


Figura 3: Skecth de la mecànica de joc

- El jugador pot canviar en qualsevol moment el color d'Owillie prement el botó del color desitjat.
- Aniran apareixent diversos elements en pantalla. Segons l'element s'haurà o bé d'esquivar o bé recollir.
- Esquivar vol dir que quan l'element arriba a la línia de moviment del personatge, Owillie no pot estar en la mateixa posició que l'element.
- Hi ha determinats elements que en comptes de ser esquivats poden ser travessats si Owillie és del mateix color que aquests elements.
- Per contra, recollir vol dir que l'element ha de coincidir en la mateixa posició que Owillie.
- L'objectiu és arribar el més lluny possible sense impactar en un dels elements que s'ha d'esquivar.

A continuació es presenta el diagrama de flux que recull totes aquestes idees.

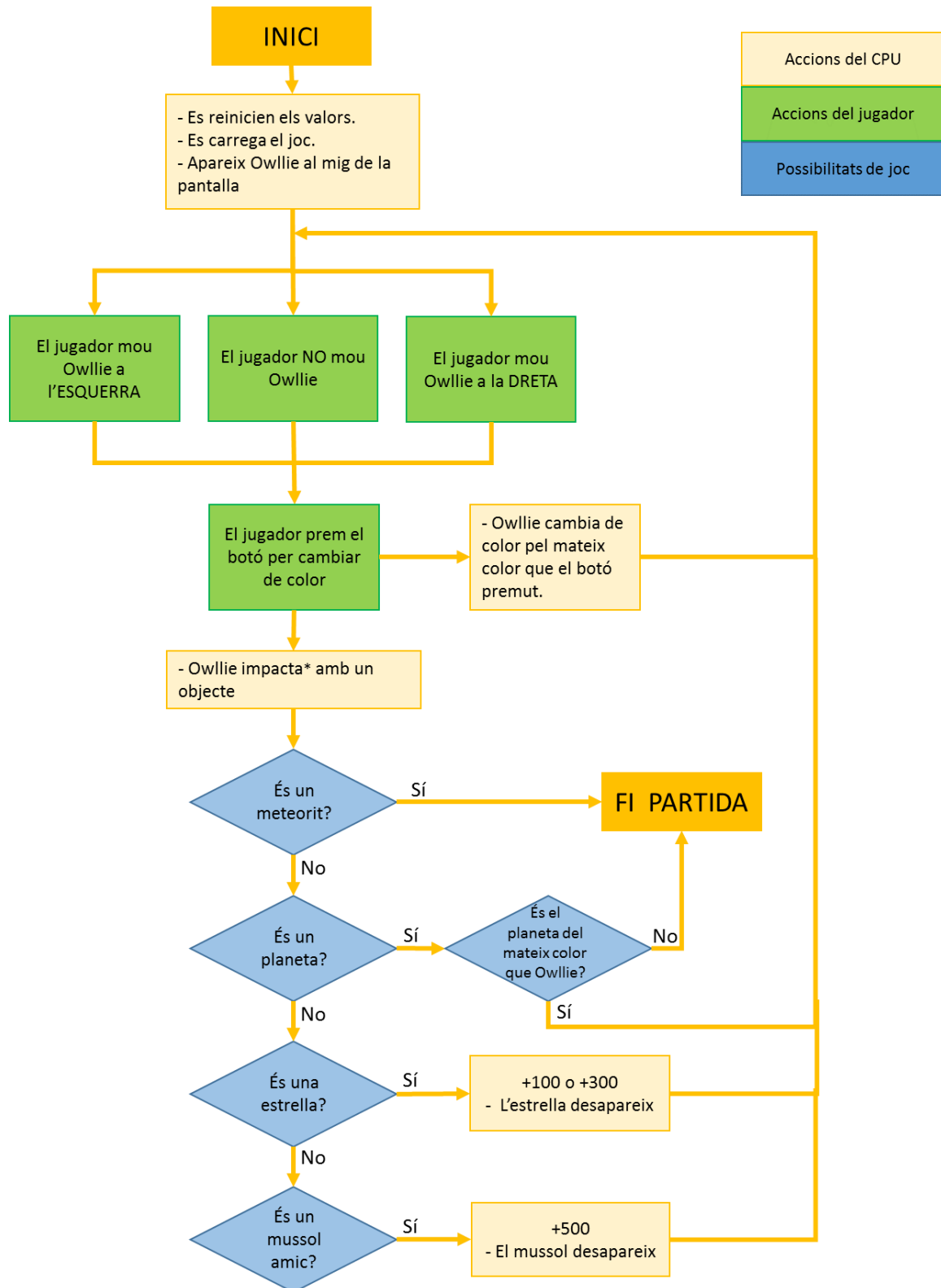


Diagrama 2: Mecànica del joc Super Owillie

Un cop finalitzada la partida es tindrà en compte el mode de joc. Segons aquest, es procedirà d'una manera o altra a fer el recompte de puntuació i, si s'escau, es desbloquejarà el següent nivell.

3. Arquitectura de l'aplicació

Definides les funcions de l'aplicació i els modes de joc, és el torn de definir les diferents pantalles que tindrà l'aplicació, com serà la navegació i quins elements es presentaran en la pantalla principal.

Al construir una *game app*, s'ha de ser clar i directe. Masses pantalles intermèdies entre la pantalla principal i el joc pot suposar una pèrdua d'interès per part de l'usuari. Així que s'ha intentat reduir al màxim el nombre de pantalles.

S'ha considerat emprar una estructura d'arquitectura jeràrquica on la pantalla més important és la pantalla d'inici anomenada *Main Menu*. A partir d'aquesta pantalla es pot arribar a qualsevol punt de l'aplicació.

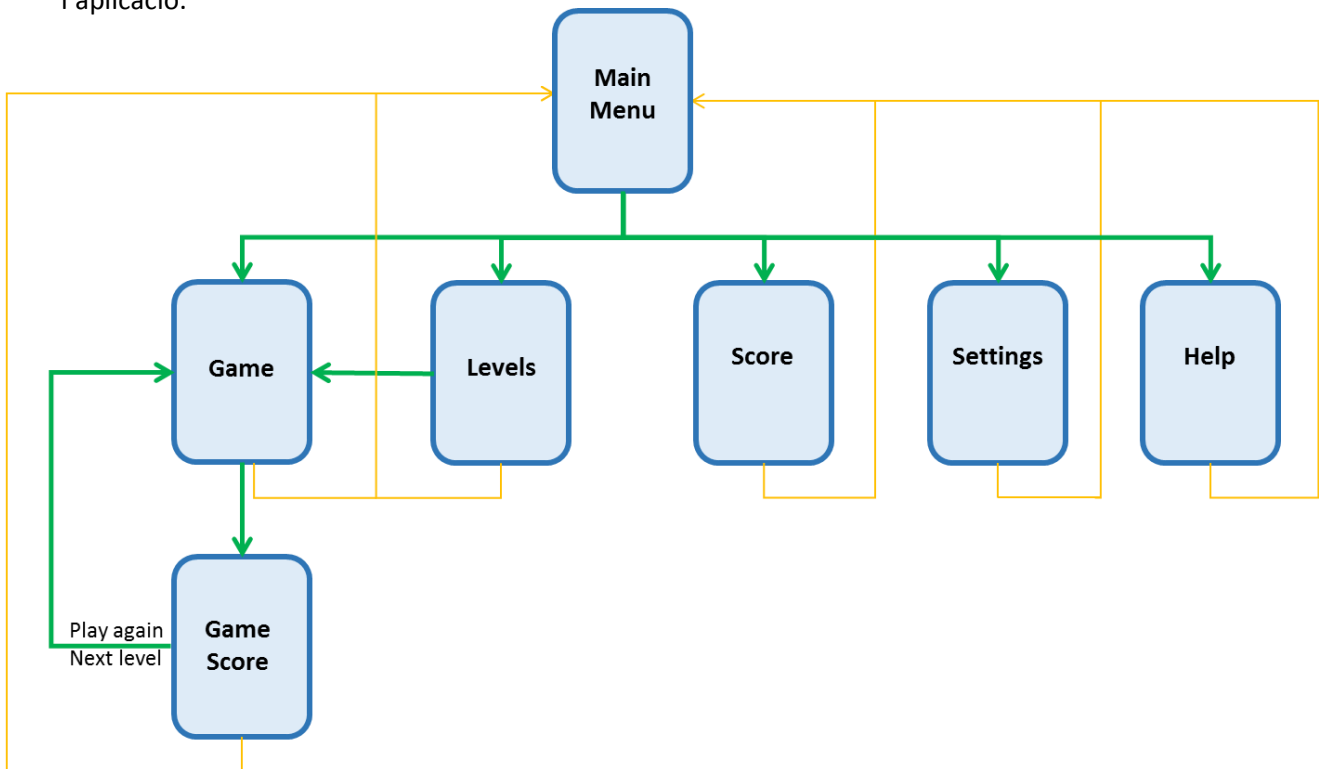


Diagrama 3: Arbre de navegació

Les diferents pantalles de l'aplicació són:

- *Main Menu*

És la pantalla principal i des d'aquesta es pot arribar a qualsevol punt de l'aplicació.

Les opcions que oferirà són:

- Clicable *Play Classic*: es va a la pantalla *game*.
- Clicable *Play Levels*: es va a la pantalla *levels* i posteriorment a *game*.

- Clicable *Score*: es va a la pantalla *score*.
- Imatge Engranatge: es va a la pantalla *settings*.
- Imatge Interrogant: es va a la pantalla *help*.

- *Levels*

Aquesta és una pantalla prèvia al joc en mode nivells que permet escollir el nivell al qual es vol jugar.

- *Game*

Aquesta és la pantalla de joc on apareixerà el mussol i els diferents elements. Segons el mode de joc es carregarà un nivell 0 per al *play classic* o un determinat nivell per al *play levels*.

El joc podrà ser pausat, moment en què apareixeran les opcions de *resume*, *restart* i *main menu*.

- *Game Score*

Al final del joc es mostrarà una pantalla amb la puntuació de la partida. Aquesta no serà una pantalla addicional sinó una pantalla que apareix dins del joc un cop s'hagi finalitzat la partida (*Game over o Level completed*).

- *Score*

Des del menú principal es podran consultar les millors puntuacions del mode clàssic.

Aquesta pantalla, i seguint l'exemple de Temple Run, mostrarà les millors puntuacions aconseguides en una única partida i les millors puntuacions aconseguides durant tot el joc.

- *Settings*

Aquesta pantalla de configuració oferirà la possibilitat d'activar o desactivar la música i els efectes sonors.

- *Help*

Una pantalla que explica les regles bàsiques del joc per si es té algun dubte.

En un primer moment la pantalla d'ajuda no estava contemplada i ha estat a partir de les proves d'usuari quan s'ha considerat oportú crear-la.

Pel que fa la navegació es podrà retornar a la pantalla principal des de totes les pantalles. Es pot veure representat per fletxes grogues en el diagrama anterior. Dins de l'aplicació, per retornar a la pantalla principal caldrà prémer el botó *main menu* que normalment es trobarà a la part inferior esquerra.

4. Disseny gràfic i interfícies

Posant en comú els apartats anteriors, és pot començar a treballar en el disseny gràfic i el tractament visual. El producte que es vol aconseguir es divideix en dues parts: el joc i l'aplicació en sí. Per això, l'apartat de disseny ha tractat aquests dos punts per separat però tenint en compte que es tracta d'un mateix producte el qual ha de seguir una mateixa línia visual.

4.1 Interfície del joc

La primera decisió de disseny important és que el joc es desenvoluparà en pantalla vertical. El motiu és que alhora de jugar-hi la pantalla vertical permet crear una interfície amb la qual interactuar amb una sola mà. A més, aquesta mà quan es mou no cobreix tant la pantalla com sí ho faria si la pantalla fos horitzontal.

Aquest fet ens determina la posició dels botons que fan que el mussol canviï de color. Aquests hauran d'estar situats en la part inferior de la pantalla. Un bon exemple de la bona funcionalitat d'aquests botons inferiors és el joc Tap Tap Ravege.



Figura 4: Captura de pantalla del joc Tap Tap Ravege – Interfície joc

Alhora, aquest fet també ens condiciona el número de colors en què el mussol podrà variar. El màxim de botons -i en el seu defecte de colors- permesos serà de 5. Si en volguéssim més, aleshores aquests tindrien un tamany més reduït i podria dificultar la interacció del joc.

Així doncs, el mussol només podrà variar entre cinc colors: vermell, groc, blau, verd i lila. Això sí, el mussol que es carrega a l'iniciar la partida estarà descolorit (color gris). Un cop se li hagi assignat un color, no tornarà a ser de color gris fins que es reiniciï la partida.

La següent figura mostra el resultat obtingut de posar en comú aquestes idees sobre l'orientació vertical i els botons que permeten controlar el color del mussol. De fet, aquesta figura mostra el que seria la pantalla *Game*, pantalla on es desenvolupa el joc.

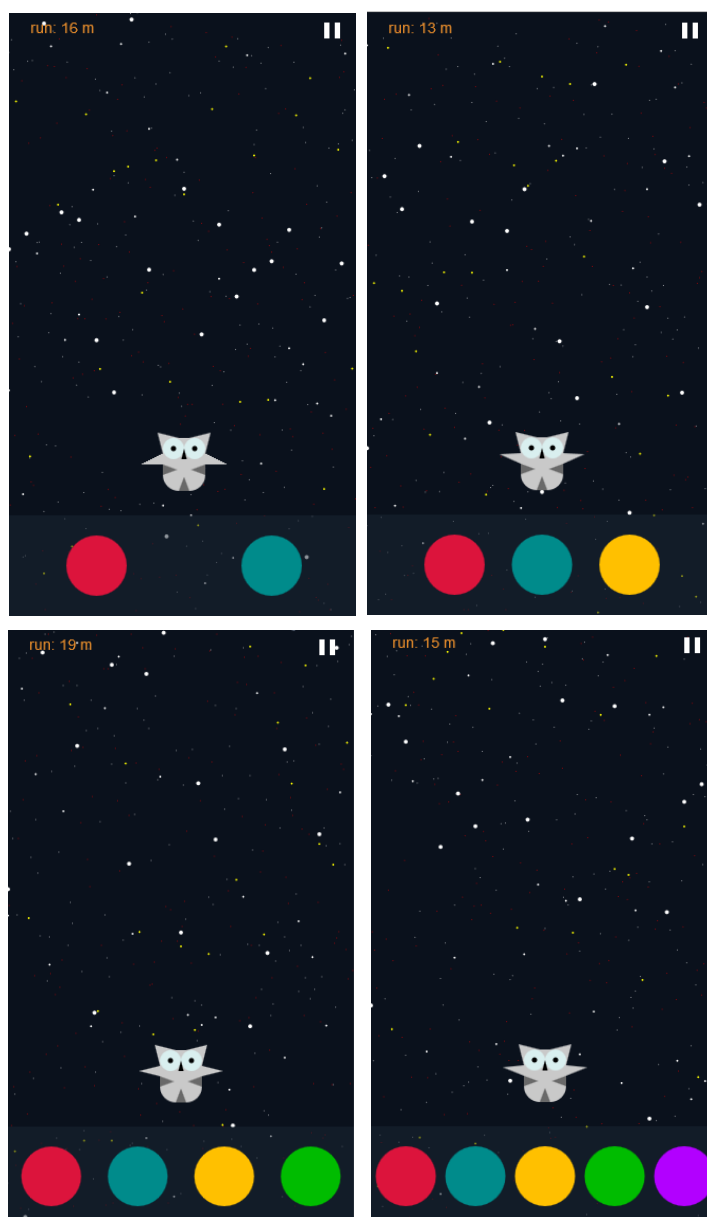


Figura 5: Captures de pantalla del joc Super Owillie – Interfície joc

Com es pot apreciar el joc compta, depenent de la dificultat, amb més o menys botons/colors. El mínim són 2 i el màxim són 5.

El joc inclou també tres pantalles addicionals: menú pausa, menú *level completed* i menú *game over*. Al menú pausa s'hi pot accedir directament des de la pantalla de joc. És el propi jugador el que decideix si vol pausar el joc, reiniciar-lo o tornar a la pantalla principal. Aquestes són les tres opcions que es mostren en aquest menú.

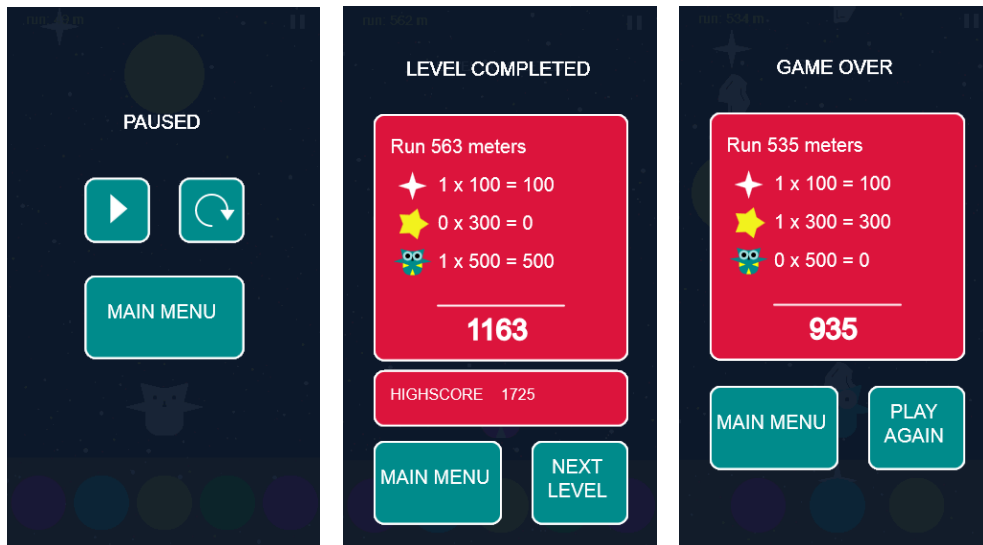


Figura 6: Captures de pantalla del joc Super Owillie - Menús

Per contra, el menús de *level completed* i *game over*, s'hi accedeix automàticament una vegada finalitzada la partida. Aquestes pantalles serveixen per mostrar la puntuació final i seguir endavant en el joc. Són dues pantalles que representen la pantalla *Game Score* (descrita en el punt 3); segons el què hagi succeït en el joc es mostrarà una o altra pantalla.

Ambdues pantalles mostren un disseny semblant que es diferencia per petits matisos:

- La pantalla de *game over* et permet tornar a jugar al nivell un cop has perdut mentre que la pantalla *level completed* et deixa passar al següent nivell (si aquest nou nivell està desbloquejat).
- A més la pantalla *level completed* guarda la puntuació més alta i la mostra.

Evidentment, des d'ambdues pantalles es pot accedir al menú principal.

4.2 Interfície de l'aplicació

L'aplicació consta bàsicament de cinc pantalles: *Main Menu*, *Levels*, *Score*, *Settings* i *Help*. Totes aquestes pantalles estan interrelacionades entre elles i es comuniquen a través d'una sèrie de botons. Per a definir tots aquests elements d'interacció ha estat de gran ajuda haver definit prèviament l'arquitectura i navegació de l'aplicació i haver elaborat un prototip en baixa definició.

Aquest prototip ha permès dissenyar com i quins elements ha d'incloure cada pantalla. A partir d'aquest document s'ha pogut definir cadascuna de les pantalles esmentades.

La pantalla d'inici inclou cinc elements d'interacció. Tres d'ells relacionats amb l'experiència del joc i els altres dos que mostren opcions secundàries. Mooniz és un joc que té com a pantalla d'inici un disseny que permet mostrar aquests dos tipus d'elements. Aquest disseny ha servit d'inspiració per al disseny de Super Owillie.

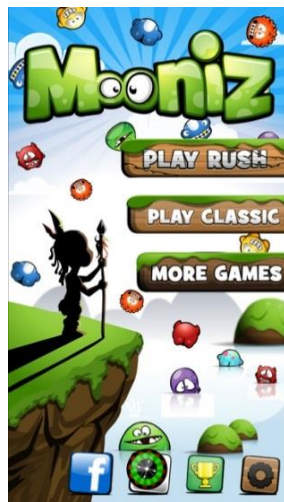


Figura 7: Captures de pantalla del joc Mooniz – Pantalla d'inici

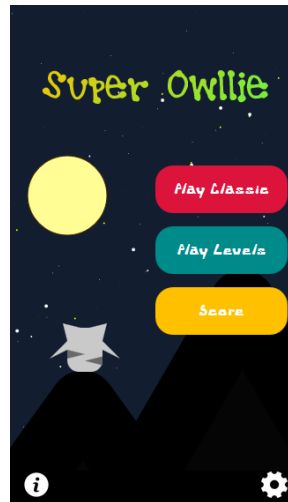
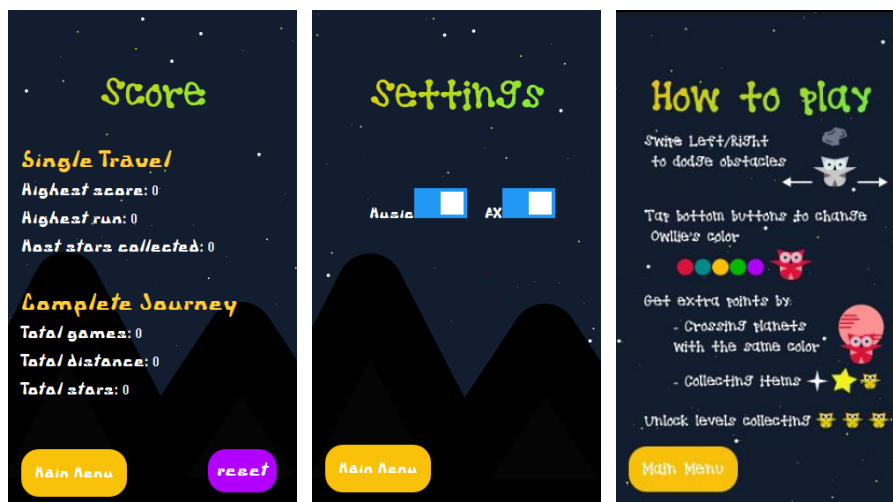


Figura 8: Captures de pantalla del joc Super Owillie – Pantalla d'inici

La resta de pantalles presenten un disseny similar: títol, cos on es troba el contingut de cada pantalla i botons de navegació. A la part inferior esquerra trobem sempre el botó de retorn a la pantalla d'inici. A la part inferior dreta no hi ha res o, si s'escau, trobem botons que realitzen altres accions.



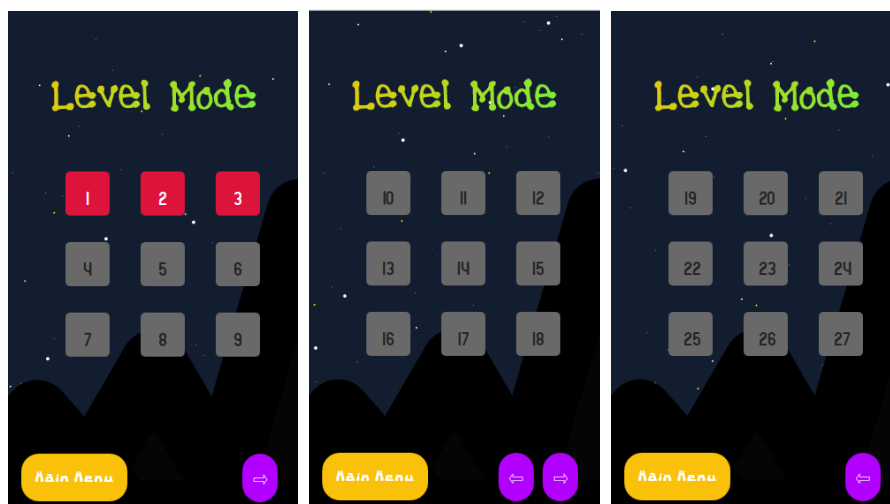


Figura 9: Captures de pantalla del joc Super Owillie – Pantalles Aplicació

El fons de les pantalles s'inspira en la història de Super Owillie, "l'últim mussol de la Terra que contempla cada nit la lluna fins el dia que decideix marxar cap a l'espai exterior". En el menú principal és pot veure aquest mussol contemplant la lluna, en les altres pantalles només queden les muntanyes ja que Owillie ha marxat cap a l'espai, finalment el joc representa Owillie durant la seva aventura còsmica.

4.3 Disseny dels elements de joc

Un dels principals reptes del projecte és crear tot el joc a partir de codi, això inclou també els diferents elements. Per això s'ha intentat dissenyar aquests elements a partir de formes bàsiques: cercle, rectangle, triangle i rectes.

A continuació es mostren els esbossos dels diferents elements juntament amb el seu resultat final:

Owillie i friendly Owls

Al personatge principal se li ha volgut donar una caracterització divertida i desenfadada. Per això, i seguint un estil Angry Birds, s'ha proposat un mussol amb uns ulls ben grossos i amb colors vius.

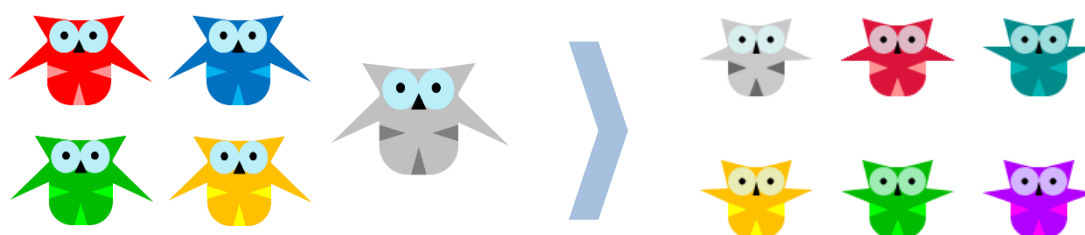


Figura 10: Disseny dels elements de joc - Owillie

Els mussols amics tindran aquesta mateixa caracterització l'únic que seran de tamany més reduït i poden variar en quant a combinació de colors. 🦉 🦉 🦉

Meteorit

Aquest ha estat l'element més complicat de dissenyar degut a la seva forma irregular. Es tracta d'una simple roca de tamany reduït.



Figura 11: Disseny dels elements de joc - Meteorit

Planetes

Per tal de donar variació als elements del joc s'ha proposat la creació de diferents planetes i colors. En total una combinació de 20 planetes diferents que apareixen en el joc de forma aleatòria. D'aquesta manera s'aconsegueix donar-li al joc més color i diversitat.

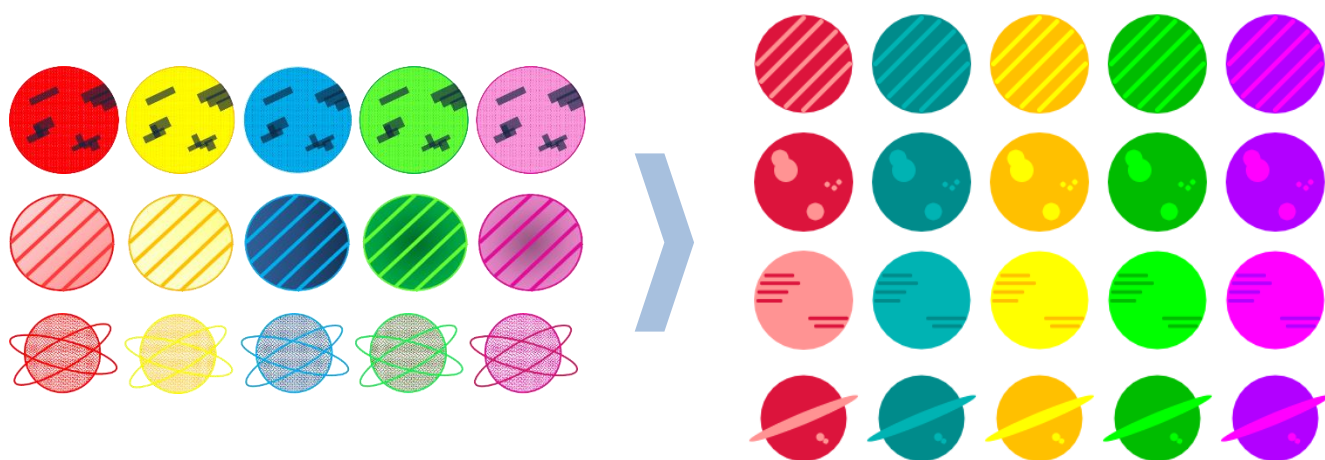


Figura 12: Disseny dels elements de joc - Planetes

Estrelles

En un principi es volia dissenyar només un tipus d'estrella. Finalment es va optar per fer-ne dos i així cadascuna tindria una puntuació diferent.



Figura 13: Disseny dels elements de joc - Estrelles

Fons

La caracterització del fons respon a dues intencions:

D'una banda crear la sensació d'univers tot recreant un ambient astronòmic juntament amb els elements anteriorment descrits. I d'altra banda, donar la sensació de moviment. En un joc d'*endless running* és important que hi hagi elements externs que no intervenen en el joc però que es mouen. Això és el que genera la sensació de que la pantalla es mou i avança.

Per tots aquests motius, el fons queda caracteritzat per un color fosc (blau molt fosc) i amb puntets de diferents grandàries i velocitats que representen estrelles.

4.4 Altres: Icones, colors i fonts

Icones

Imatge de l'aplicació que es generarà a l'escriptori del *Smartphone*.



Figura 14: Icona de l'aplicació Super Owillie

Colors

Els principals colors usats en l'aplicació són:

Aux blanc	(255, 255, 255)	#ffffff	Estrelles de 4 arestes, botó pausa, partícules del fons, text,
Aux negre	(0, 0, 0)	#000000	Owillie, <i>friendly owl</i> , meteorit, muntanyes, números <i>locked levels</i>
Aux vermell	(255, 0, 0)	#ff0000	Partícules del fons
Aux groc	(255, 255, 0)	#ffff00	Partícules del fons
Fons 1	(18, 30, 48)	#121e30	App <i>background</i>
Fons 2	(10, 17, 28)	#0a111c	Game <i>background</i>
gris 1	(201, 201, 201)	#c9c9c9	Owillie, meteorit
gris 2	(105, 105, 105)	#696969	Owillie, botons <i>locked</i>

vermell 1	(220, 20, 60)	#DC143C	Owillie, planetes, menú puntuacions, botons, botons <i>unlocked</i> ,
vermell 2	(255, 147, 147)	#ff9393	Owillie, planetes
blau 1	(0, 139, 139)	#008B8B	Owillie, planetes, botons, botons joc, fons menú puntuacions
blau 2	(0, 179, 179)	#00b3b3	Owillie, planetes
groc 1	(255, 192, 0)	#ffc000	Owillie, <i>friendly owl</i> , planetes, text, botons, botó <i>Main Menu</i> ,
groc 2	(255, 255, 0)	#ffff00	Owillie, planetes, estrelles 5 arestes
verd 1	(0, 188, 0)	#00bc00	Owillie, <i>friendly owl</i> , planetes
verd 2	(0, 255, 0)	#00ff00	Owillie, planetes
lila 1	(178, 0, 255)	#b200ff	Owillie, planetes, botons
lila 2	(255, 0, 255)	#ff00ff	Owillie, planetes

Fonts

L'aplicació usa principalment tres fonts.

- Títols: brookeshappelldots. Degradat groc-verd. 50px.
- Botons: brookeshappelldots. Blanc. 20px.
- Text: brookeshappelldots. Blanc i Taronja. 16px i 30px.
- Números: Quicksand. Blanc i negre. 16px i 22px.
- Joc: (*default*) no se li ha assignat una tipografia especial.

El veloz murcilafo
1234567890

Figura 15: Exemples tipogràfics

Totes les fonts descrites s'han descarregat d'internet i tenen una llicència no comercial *creative commons*, o similar, la qual permet fer-ne ús gratuït i incrustar-la a l'aplicació.

Capítol 4: Implementació

Dins del projecte, l'etapa d'implementació ha suposat la de major esforç. Aquesta part del projecte engloba tres tasques principals:

- **Implementació de l'app:** consisteix en la implementació les diverses pantalles (cada pantalla equival a una pàgina html) i s'enllacen entre elles de manera que es crea una aplicació totalment navegable.
- **Implementació del joc. Elements:** consisteix en la implementació de cadascun dels elements que apareixeran durant el joc – *Owl, Planets, Meteor, Stars, Black Hole*, el fons i els controls (botó pausa i botons que permeten canviar de color el mussol).
- **Implementació del joc. Algorísmica:** consisteix en la implementació de la lògica del joc, és a dir, totes aquelles funcions que fan que el joc es desenvolupi correctament. Addicionalment i dins d'aquest apartat, també s'han dissenyat les funcions que s'utilitzen en l'aplicació (modes de joc, *Score, Settings...*).

Aquestes tasques estan relacionades entre sí i per això no poden ser tractades per separat. Per això en els següents apartats es tractarà la implementació global de l'aplicació.

1. Metodologia

El codi s'ha escrit usant Adobe Dreamweaver i s'han emprat diversos llenguatges web: html, css i javascript. Dins de javascript s'ha optat per usar la biblioteca de programació visual P5.js perquè aquesta ofereix la possibilitat d'usar el canvas i, a diferència de javascript, presenta funcions bàsiques de dibuix més senzilles d'usar com `ellipse()` per crear un cercle o `rect()` per crear un quadrat. D'aquesta manera la part gràfica ha resultat menys laboriosa que usant exclusivament javascript.

La carpeta que conté el codi s'estructura en diversos directoris:

- `assets`: inclou imatges de suport que s'han de carregar.
- `css`: inclou dos fulls d'estil. Un per a l'html i l'altre per als elements d'script.
- `font`: inclou les diferents tipografies baixades d'internet (Royalty Free).
- `html`: inclou les diverses pàgines de l'aplicació (.html).
- `js`: inclou els arxius de javascript que són els que contenen les funcions de l'algorísmica del joc i aplicació.
- `llibreries`: conté les llibreries de P5.js referent a processing, manipulació del DOM i sons.
- `sounds`: conté les músiques i efectes sonors baixats d'internet (Royalty Free).
- `game.html`: respon a la pàgina html del joc.
- `index.html`: respon a la pàgina d'inici o menú d'inici de l'aplicació.

En la carpeta js es troben tota una sèrie d'arxius. Això és debut a què s'ha optat per dividir l'algorísmica en múltiples arxius javascript. Principalment perquè això ha contribuït a tenir un codi més estructurat. Al cap i a la fi es tracta d'un codi llarg amb moltes funcions i de vegades pot resultar complex trobar una determinada funció. La divisió de codi s'ha fet de forma temàtica en variables, formes, classes, elements, funcions i background (diferents fons dependent de la pantalla).

Un altre motiu, i aquest a nivell funcional, per separar el codi és perquè d'aquesta manera es pot determinar manualment quina pàgina html té accés a quina part de codi. Per exemple, la pàgina d'Score té accés a les variables i per això pot accedir a la informació de les puntuacions. Es podria dir que d'aquesta manera es crea un link entre una pàgina html i el propi joc.

Els arxius .js més importants i que cal considerar són:

- **shapes:** inclou les funcions que dibuixen cadascun dels elements - *Owillie, Planets, Meteor, Stars, pause button* i els controls per canviar el color del mussol.
- **elements:** una llista creada manualment que inclou cada element en les seves tres posicions. Aquesta llista és la que serveix com a referent per crear la llista d'elements de cada nivell. La funció "addElementTo()" escull aleatòriament un element d'aquesta llista i el copia en una altra llista que és la que es carregarà en el joc. D'aquesta manera s'ha aconseguit crear nivells aleatoris i, a més, que un mateix nivell sempre sigui diferent.
- **variables:** és la part de codi que recull les diferents variables. Addicionalment s'han dissenyat també les funcions que són cridades des de les altres pàgines de l'aplicació, ja que es tracta de setters i getters de les variables tot usant el localStorage().
- **functions:** podríem dir que és l'arxiu més important ja que conté tota l'algorísmica del joc. Aquí és on s'han dissenyat totes les funcions que defineixen el joc i el fan funcionar.

Cal considerar que de vegades hi ha funcions que pertanyen a dos arxius diferents, en aquests casos s'ha valorat quina és la finalitat i a partir d'aquí s'ha decidit un arxiu o altre.

Respecte a l'HTML i el CSS, aquests no presenten una gran complexitat pel que fa el codi així que en aquest capítol no es comentaran explícitament.

2. Resum de la lògica del joc

La lògica del joc està implementada a partir de P5.js que és una llibreria de javascript per a la programació visual que segueix les bases de Processing.

Aquesta llibreria presenta una sèrie de funcions estructurals que s'usen per a definir l'esquelet del programa. D'aquestes funcions estructurals, les més importants i les que s'han usat en el codi són: `preload()`, `setup()` i `draw()`.

- La funció `preload()` serveix per carregar abans de què s'executi el joc certs arxius com les músiques.
- `Setup()` crea el canvas i en defineix l'entorn.
- `Draw()` executa contínuament les línies de codi contingudes dintre d'aquesta funció fins a què el programa para.

El cos del joc està contingut dintre de la funció `draw()`. En primer lloc verifica a partir de les variables `gameover` i `paused` si la partida ha acabat o si el joc està pausat. En aquest cas, el programa s'atura i es mostren els menús corresponents.

En cas que aquestes variables siguin falses, s'executen les línies de codi que creen el joc. El funcionament és simple, en cada interval de la funció `draw()` es redibuixa el canvas amb la diferència que els elements tenen una nova posició. L'efecte òptic és el de moviment dels elements. En cinematogràfica un segon de vídeo equival a 25 *frames* o 25 imatges que es visualitzen en un segon de temps. Això vol dir que una sèrie d'imatges immòbils donen la impressió de moviment si es visualitzen ràpidament. Precisament la funció `draw()` fa això, redibuixa a gran velocitat les imatges que hi ha en el canvas, concretament 60 imatges per segon.

Dins de la funció `draw()` es criden altres funcions, la més important és `uploadElements()`. Aquesta funció és la que s'encarrega de crear els elements que apareixen per pantalla, posicionar-los i actualitzar-ne la posició.

Una de les decisions més importants a l'hora de programar ha estat determinar com es crearien els nivells. La primera opció era crear una llista d'elements fixa que es carregaria al principi del nivell. La segona opció era fer que aquesta llista que es carrega fos aleatòria. A pesar de la dificultat, aquesta segona opció era la millor i, per tant, es va procedir a crear una llista d'elements aleatòria.

uploadElements() crea la llista d'elements cada vegada que el joc s'inicia/reinicia. El seu funcionament és senzill: hi ha dues llistes listof_elements i levels.

- La primera llista és un llista d'objectes que ja està definida i no es pot modificar. La llista la formen els diferents elements del joc (meteorits, planetes, estrelles...) i conté totes les possibles posicions per a cada element (pos1, pos2 i pos3).

Per exemple conté un element meteorit en la posició 1, un altre en la posició 2 i un altre en la posició 3; després un planeta de color vermell en la posició 1, un altre en la 2 i un altre en la 3; i així amb els 9 elements existents.

Cada objecte de la llista té una sèrie de propietats: un identificador (id), posició en l'eix de la x (x), posició en l'eix de la y (y), un estatus (status) i altres propietats que depenen del tipus d'element.

Cal dir que la llista incorpora elements repetits com són els meteorits i les estrelles de quatre arestes. Això s'ha definit així perquè es vol que aquests elements en concret tinguin més opcions de ser seleccionats, se'n vol augmentar la probabilitat.

- La segona llista està buida en el moment de començar el nivell. Quan es crida a la funció uploadElements() aquesta llista levels s'omple amb elements de la primera llista. El que es fa és que aleatòriament s'escull un nombre entre 0 i 51 i aquest nombre serà la posició de la llista listof_levels[nombre aleatori]. D'aquesta manera escollim un element de la llista listof_levels que copiem a la llista levels.

S'obté així una llista levels diferent i totalment aleatòria cada vegada que es carrega el nivell.

Dos aclariments. En aquest punt és el nivell el que determina quants elements s'han de copiar a la llista (nivell 1, 50 elements; nivell 2, 100 elements) i en el moment en què es copia un planeta, se li assigna també un tipus de manera que el dibuix del planeta varia entre quatre possibilitats.

Un cop elaborada aquesta llista, només es realitza una vegada tot i que constantment s'estigui cridant a la funció uploadElements(). El que fa uploadElements() és recórrer aquesta nova llista levels i actualitzar-ne la posició "y" en cada iteració de draw(). Això és el que fa que visualment veiem que els elements van descendent per la pantalla.

Finalment uploadElements() crida a una sèrie de funcions que el que fan és afegir propietats i controlar si el mussol col·lideix amb els elements de la llista. En cas de col·lisió es mira quin és l'identificador de l'element amb el qual s'ha impactat i si es tracta:

- d'un meteorit el valor gameover passa a ser vertader.

- d'un planeta se'n verifica el color del mussol i segons si presenta el mateix color o no el valor `gameover` es deixa en fals o es canvia a vertader.
- d'una estrella o un mussol amic aquest element canvia el seu estat (status) a fals i desapareixerà de la pantalla.

La funció `draw()` a més dibuixa el personatge principal, un mussol que se situa a la part inferior del canvas. Aquest mussol només es pot moure a dreta o esquerra i aquest moviment queda controlat per la funció `touchEnded()`.

Com que el mussol pot canviar de color, s'han introduït una sèrie d'esdeveniments. S'han creat una sèrie de botons clicables que permeten que el mussol canviï de color tot cridant a la funció adequada.

Aquesta part de manipulació de botons s'ha aconseguit a partir de la biblioteca `p5.dom` que és una biblioteca específica de P5 que permet la interacció amb els elements HTML tot manipulant el DOM. En aquest sentit s'ha pogut crear etiquetes `<button>` i assignar-li una classe o id que ha permès la modificació de l'estil en el CSS.

Finalment el darrer aspecte per comentar és el tema de la connexió aplicació-joc, molt relacionat amb les puntuacions. L'aplicació compta amb un seguit de pantalles que són individuals les unes a les altres. Tot i així el que succeeix dintre de la pantalla del joc és rellevant per a les pantalles: *score* i *levels*.

Score mostra les millors puntuacions del joc. *Levels* mostra nivells bloquejats o desbloquejats en funció de la quantitat de petits mussols que s'han recollit.

Per aconseguir que les dades obtingudes en el joc estiguessin disponibles al llarg de les diferents pantalles de l'aplicació, el que s'ha hagut d'implementar són variables que guarden el seu valor localment en el navegador de l'usuari. Això és el que fa `localStorage()` i presenta un millor resultat que les *cookies*.

Per això tot el tema de puntuacions dins del joc fa ús d'aquesta API d'HTML.

3. Dificultats trobades

Les principals dificultats que s'han trobat al llarg de la implementació són:

- Dificultats a l'hora d'implementar. Algunes funcions han estat complicades de dissenyar. Ha calgut paciència i anar provant. Al final s'ha aconseguit que l'aplicació faci tot el què es pretenia.
- La implementació inicial partia d'una sèrie de funcions bàsiques. A mesura que s'anava avançant sorgia la necessitat d'implementar més i més funcions. Aquest és un dels motius pel qual s'ha allargat l'etapa d'implementació.
- Definir moviments tàctils ha resultat complicat ja que la documentació que hi ha sobre les funcions tàctils en P5.js és escassa. Al final després de nombroses proves i intents es va aconseguir implementar-ho. El gran referent va ser la funció `mouseDragged()` a partir de la qual es va intentar simular i adaptar amb els elements tàctils.
- Obtenir les puntuacions finals també ha estat una tasca laboriosa. Com que el canvas no s'atura i va dibuixant *frame a frame* sense parar, de vegades és difícil obtenir el valor final de les variables. Per això s'han hagut de crear variables que han ajudat a que les funcions es realitzessin només una vegada.
- L'aplicació ha estat creada a partir de diverses pàgines html totes concatenades entre sí. El problema que ha sorgit ha estat com aconseguir que l'actualització dels valors de les variables d'una pàgina a una altra. Això s'ha solucionat emprant les funcions d'HTML5 `sessionStorage()` i `localStorage()`.
- Ja que s'ha pretès que el canvas tingués unes dimensions idèntiques a les dimensions de la pantalla del dispositiu, s'ha hagut de tractar tot el posicionament dels elements, formes dels menús de i botons de forma relativa. Aquesta ha estat una tasca complexa ja que algunes funcions (sobretot les primeres de totes) s'han hagut de retocar. Al final el que s'ha fet és crear una graella de 10x10 i posicionar els elements segons les noves coordenades de la graella. Per exemple per situar un objecte al mig del canvas seria $(w*0.5, h*0.5)$.
- Durant l'adaptació a diferents dispositius s'han hagut d'implementar algunes línies de codi pertanyent a Android/iOs per tal d'anular el que ve per defecte. En alguns casos s'ha hagut d'investigar i això ha comportat temps. Un exemple és la configuració del `<viewport>` per tal d'eliminar la barra de direccions del navegador del mòbil.
- A l'hora d'allotjar el joc en el servidor, han aparegut els primers errors d'incompatibilitat. El codi funcionava bé en els navegadors de l'ordinador, però no en els navegadors dels mòbils (sobretot en iOS). Ha calgut revisar tot el codi per trobar petits errors en l'html i javascript que impedièen el bon funcionament.
- Un altre inconvenient ha estat aconseguir l'apk, doncs el programa que inicialment havia escollit Intel XDK no permetia encapsular el codi a no ser que es tinguessin els certificats de desenvolupador y *keystores* d'Android. Finalment s'ha decidit presentar el producte en format web app ja que no ha donat temps d'aprendre a usar Android Studio que hauria estat l'alternativa a Intel XDK.

4. Funcions implementades

SHAPES	
Funció	Descripció
drawOwl (x, y, maincolor, decorcolor)	Dibuixa el mussol protagonista, Owillie. El color per defecte és gris.
drawMiniOwl (x, y, maincolor, decorcolor)	Mateixa funció que l'anterior però amb diferent tamany i diferents valors per a x i y. Aquesta funció serveix per dibuixar els friendlyOwls durant el joc i el mussol que apareix en la pantalla de puntuacions.
planet1(x,y,maincolor,decorcolor)	Dibuixa el planeta del tipus 1, amb línies diagonals.
planet2(x,y,maincolor,decorcolor)	Dibuixa el planeta del tipus 2, amb rodones
planet3(x,y,maincolor,decorcolor)	Dibuixa el planeta del tipus, amb línies horitzontals
planet4(x,y,maincolor,decorcolor)	Dibuixa el planeta del tipus, amb anell. <i>[planeta de menor radi]</i>
meteor(x,y)	Dibuixa el meteorit. Aquest s'ha dibuixat a partir de línies rectes que s'uneixen entre elles i creen una estructura deforme.
meteorSpin(x,y)	Fa girar el meteorit.
star(x, y, npoints)	Dibuixa les estrelles depenent del número d'arestes. En el joc seran de 4 arestes o 5. <i>[Aquesta funció ha estat extreta de: https://p5js.org/examples/form-star.html sota una llicència CC Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)]</i>
starSpin(x, y, npoints)	Fa girar les estrelles. En el joc només girarà l'estrella de 5 arestes.
miniStar(x, y, npoints)	Mateixa funció que crea les estrelles però aquesta vegada són estrelles de tamany reduït que ens serviran per a la pantalla de puntuacions, al finalitzar la partida.
<i>blackhole(x, y)</i>	<i>Dibuixa el forat negre.</i> <i>[Aquest objecte ha estat suprimit]</i>
finalElement(y)	Dibuixa l'element que indica que el nivell ha acabat. Aquest consisteix en un rectangle que ocupa tota la pantalla i que canvia tant ràpid de color que dona un cert efecte òptic (el qual recorda al SuperMario quan ha recollit l'estrella i és invencible).
Particle(_pX, _pY, dir, color, r)	Classe que defineix partícules aleatòries. Aquestes partícules simularan les estrelles de fons.

backgroundStars()	Dibuixa les partícules que s'han creat en la funció anterior i les mostra en el canvas. La particularitat d'aquesta funció és que mostra les partícules en tres tamanyes i velocitats diferents. D'aquesta manera, s'aconsegueix obtenir una certa profunditat i moviment que afavoreixen al crear l'entorn còsmic del joc.
controlers(num)	Donat un cert número, es dibuixen en el canvas el número de botons (amb els seus respectius colors) que permeten canviar el color del mussol. <i>[El canvi de color es tracta en una altra funció]</i>
pausedIcon()	Dibuixa l'icona de pausa al marge superior dret.
loadingAnimation()	Crea una animació que apareix durant el temps de <i>loading</i> .

FUNCTIONS	
Funció	Descripció
Funcions estructurals	
Preload()	Funció que carrega els arxius abans de configurar el canvas.
Setup()	Funció que configura el canvas.
Draw()	Funció que dibuixa en el canvas per cada iteració.
Funcions per crear la llista de cada nivell	
addElementTo()	afegeix un element aleatori de la llista blocs_elements a la llista "level". Y assigna una posició en l'eix Y a cada element. <i>[La distància entre element i element és de 50px. Cosa que s'hauria de millorar.]</i>
controllingMeteor()	Arregla la llista per evitar que hi hagi línies de meteorits impossibles. Bàsicament el que fa és controlar que no hi hagi 4 meteorits seguits i en cas que n'hi hagi, en canvia uns quants per estrelles de quatre arestes. El procediment és canviar l'id de l'element, de id=3 a id=4.
positioningElements()	Corregeix la llista d'elements per evitar que aquests se solapin. Considera el diàmetre/llargària de cada element i segons l'id de l'element i l'id de l'element anterior determina una distància o altra.
uploadElements()	Es completa la llista "level" amb la introducció de l'element final i 3 friendlyOwls. A continuació es recorre la llista i es dibuixa cadascun dels elements segons la seva propietat id. Id=1 Planet Id=2 BlackHole <i>[no usat]</i>

	<p>Id=3 Meteor</p> <p>Id=4 Star (4 arestes)</p> <p>Id=5 Star (5 arestes)</p> <p>Id=0 Friendly Owl</p> <p>Id=13 Final Element</p> <p>També actualitza en cada interacció de draw(), la posició en l'eix de la Y dels elements. De manera que es van apropant cap al mussol.</p>
Funcions per controlar diversos successos	
setStatus(element)	<p>Si l'element es troba fora del canvas el seu staus és false. Això és així perquè draw() només dibuixa els elements que es mostren en el canvas la resta d'elements no té sentit dibuixar-los fins a que apareguin pel canvas. D'aquesta manera renderitzem el joc.</p> <p>Si tenim una llista de 1000 elements i els dibuixem tots alhora, el joc anirà extremadament lent. Com que només ens interessa els elements que són visibles en el canvas, té sentit no dibuixar la resta.</p>
setImpactArea(element)	<p>Defineix l'àrea on els elements poden ser tocats pel mussol. Quan un element es troba en aquesta àrea, es crea una propietat anomenada impactarea amb valor true. Quan se surt d'aquesta area la propietat passa a tenir el valor false.</p>
owlCrashesElement(element)	<p>Comprova la col·lisió entre Owillie i un element.</p> <p>Si Owillie es troba en la mateixa posició que un element amb impactarea=true, aleshores s'evalua quin tipus d'element és (segons el seu id) i es procedeix segons aquest.</p> <ul style="list-style-type: none"> - Si es tracta d'un planeta de diferent color que Owillie o un meteorit, la propietat gameover pren el valor true. - Si es tracta d'una estrella, la propietat status pren el valor false amb la qual cosa desapareix l'element de la pantalla. També se li afegeix una nova propietat, toched, que servirà per a fer el recompte de punts al final de la partida.
Funcions per controlar els colors del mussol	
OwlChangesColor(maincolor, decorcolor)	<p>Les variables owlColorM i owlColorD prenen valors nous, de manera que en la següent iteració, Owillie serà dibuixat de diferent color.</p>
OwlChangesIntoRED()	<p>Aquestes funcions són idèntiques l'únic que canvia són els colors, tant el principal com el de decoració.</p> <p>El motiu pel qual hi ha 5 funcions idèntiques en comptes d'una única funció amb un paràmetre que n'indiqui el cas (per exemple: OwlChangesInto(red)) és debut a què en la creació</p>
OwlChangesIntoBLUE()	
OwlChangesIntoYELLOW()	
OwlChangesIntoGREEN()	
OwlChangesIntoPURPLE()	

	de botons i posterior assignació d'esdeveniments, no permet passar-li funcions amb paràmetres. Per això s'han hagut d'implementar cinc funcions diferents, una per a cada cas.
Funcions que defineixen la interacció del jugador amb el joc	
keyPressed()	Owwlie es mou emprant les fletxes ← i →. En principi aquesta funció no és necessària ja que l'aplicació està dissenyada per a pantalles tàctils. Debut a què s'està treballant en un ordinador, resulta més fàcil usar les fletxes i també la he deixat per a facilitar la correcció i avaluació del projecte.
touchStarted()	Aquestes tres funcions són les que configuren els controls tàctils. És a dir el swipe del mussol. <ul style="list-style-type: none"> - touchStarted() recull el primer punt que es toca. - touchedEnded() recull el darrer punt que es toca. És a dir, el punt on el dit deixa la pantalla. - touchMoved() defineix que fa el mussol segons la comparació entre els valors obtinguts a touchStarted() i touchEnded() <i>[Comentar que funcionen correctament ja que dispo d'un ordinador amb pantalla tàctil i n'he pogut comprovar la funcionalitat]</i>
touchEnded()	
touchMoved()	
colorButtons(num)	Crea els botons que permeten fer canviar el color del mussol. Hem explicat que la funció controlers(num) dibuixava els diferents botons però que aquests no eren clicables ja que només eren simples dibuixos. Aquesta nova funció, colorButtons(num), és la que crea els diferents botons clicables al voltant de cadascun dels dibuixos. El paràmetre num indica quants botons s'han de crear i segons el nombre estaran situats en una posició o una altra i seran d'un color o un altre. Tot això queda configurat en aquesta funció.
hidemButtonsGame(num)	Els botons queden desactivats. Aquesta funció s'utilitza sobretot en el menú de pausa, fi de nivell o pantalla de game over. El motiu és que si segueixen operatius, es pot canviar el color del mussol.
Funcions de joc	
pauseGame()	Pausa el joc. Si la variable paused pren el valor true, el joc es para. Això vol dir que el canvas s'atura i ja no hi ha més interaccions.
pauseMenu()	Crea el menu de pausa amb les opcions: continue, restart, main menu.

restartGame()	Reseteja el joc i torna a carregar el nivell. En aquesta funció les variables i, sobretot, les variables de control prenen els seus valors inicials.
setNextLevel()	Carrega el següent nivell.
gameoverMenu()	Crea el menu de game over o el de level completed amb les opcions: play again/next level i main menu. A més mostra un recuadre amb les puntuacions.
Funcions per a les puntuacions i emmagatzemar dades al localStorage()	
showRun()	Mostra al marge superior esquerre la distancia recorreguda al llarg de la partida.
starTouched()	Fa el recompte del nombre d'estrelles i mussols recol·lectats.
updateHighscores()	Emmagatzema al localStorage() la millor puntuació per del nivell.
updateOwls()	Emmagatzema al localStorage() el nombre de mussols recol·lectats per del nivell. El màxim són 3 i aquesta informació ens resultarà útil alhora de desbloquejar els diferents nivells.
updateScoreScreen()	Emmagatzema al localStorage() totes les puntuacions necessàries per a la pantalla Score.

VARIABLES	
Funció	Descripció
Funcions per tal de recollir les dades del joc i usar-les en l'aplicació – Manipulació del DOM	
setinicialitzarLevel(num)	S'executa al fer click sobre un dels nivells. El que fa es carregar el nivell segons el paràmetre d'entrada, num. Cada num correspon a una posició en la llista que conté la informació sobre cada nivell. <pre>{level:21, TotalLevelElements:160, speed:-2.1, numOwlColors:5, status: "unactive"},</pre> A partir d'aquesta informació es configura el nivell. Val a dir que el nivell 0, correspon al mode clàssic i aquest es carrega al fer clic sobre el botó "Play Classic" del menú principal.
setScoreScreen()	La pantalla Score recull exclusivament les puntuacions del joc en mode Classic. En cas d'haver ja realitzat una partida en aquest mode, aquesta funció actualitza les dades.

resetScores()	<p>Aquesta funció l'he creada per tal de poder buidar la memòria del localStorage() i així fer proves. L'he deixat operativa perquè alhora d'avaluar pot ser útil.</p> <p><i>[Malahuradament té un error i és que al buidar tota la memòria també esborra el nombre de mussols recol·lectats i per tant esborra el progrés dels nivells desbloquejats.]</i></p>
setSliders()	<p>Recorda la posició dels sliders ON/OFF segons si la música està en mode ON o OFF, d'aquesta manera al carregar la pàgina de Settings els sliders prenent la correcta forma.</p>
setMusic()	<p>Es controla si l'usuari vol la música ON o OFF.</p>
setFX()	<p>Es controla si l'usuari vol els efectes sonors ON o OFF.</p>
setLockedLevels(x,y)	<p>Al carregar les pàgines del mode Levels, determina quins nivells estan bloquejats i quins no. Si estan bloquejats li assigna la classe "bLevelU" que enfosqueix el recuadre i n'elimina el link al joc. Així que si es prem el botó no succeeix res.</p>

Taula 7: Funcions implementades

5. Bugs i característiques futures

Finalitzada l'aplicació s'han realitzat les corresponents proves d'usuari. En aquestes s'han detectat una sèrie de *bugs* que s'han intentat solucionar.

Bugs solucionats

- El mussol podia variar la seva posició quan el joc estava pausat.
- S'ha arreglat el posicionament dels elements. Abans se solapaven.
- El so no funcionava quan el mussol recol·lectava un objecte.
- El menú *game over* no es dibuixava correctament.
- Al completar la partida l'opció "*Next Level*" permetia sempre passar al següent nivell, independentment si el nivell estava bloquejat o no.
- Alguns botons no responien al fer-hi clic perquè estaven mal posicionats.
- S'ha millorat el moviment *swipe*. Al canviar de color el mussol movia la seva posició degut a la sensibilitat del Touch. Ara el mussol només es mou si el *swipe* es realitza per sobre dels controls.
- Errors en l'html.
- Les fonts escollides no es visualitzaven en els dispositius mòbils. S'han hagut de buscar noves fonts.
- Les puntuacions no comptaven bé les estrelles.
- Durant l'adaptació a diversos dispositius, quan aquests presentaven una pantalla de major resolució, la col·lisió no es detectava correctament.
- Algunes músiques no es carregaven del tot. S'ha hagut de canviar el format de .mp3 a .wav ja que aquest darrer és el format més compatible amb els navegadors.

Bugs no solucionats

- La música del joc quan s'acaba no torna a començar.
- De vegades s'impacta amb un meteorit (per la part superior) i tot i activar-se el so de col·lisió el joc segueix i no es canvia l'estat a gameover.
- Pot ser que al principi de tot apareguin 4 meteorits impossibles de superar. Això només pot passar al començament perquè s'ha aconseguit eliminar quan això succeïa durant la partida.
- Al completar el nivell, el recompte de mussols es fa posterior a l'aparició del menú "*Level Completed*" amb la qual cosa pot ser que aparegui el botó "*Play Again*" quan hauria de ser "*Next Level*".
- Al fer clic en *restart* es resetegen tots els valors emmagatzemats en el localStorage, incloent-li els mussols recol·lectats. Això suposa que es perden els nivells desbloquejats.

Adicionalment, i com a resultat de les proves d'usuari, s'ha creat una pàgina d'instruccions per a ensenyar com s'hi juga i s'ha fet que travessar planetes doni punts per així motivar al jugador.

D'altra banda hi ha una sèrie de funcionalitats que no s'han implementat però que resulten interessants fer-ne la implementació en futures actualitzacions/versions de l'aplicació.

Característiques futures

- En el mode clàssic, a mesura que augmenta el temps, s'hauria d'augmentar la velocitat del joc.
- Es podria implementar una petita animació quan el mussol impacta amb un element i abans de la pantalla "GameOver". Els ulls podrien canviar en comptes dels puntets negres, aquests es podrien convertir en creus.
- En la pantalla *Levels*, portar un control visual del nombre de mussols recol·lectats per nivell. La idea seria fer quelcom similar a Angry Birds.



Figura 16: Captura de pantalla del joc Angry Birds - Nivells

- Afegir l'element del *BlackHole* que seria com un forat negre que al entrar-hi s'aniria a una pantalla diferent només amb estrelles. (semblant a la canonada verda de Mario)
- Afegir *power ups*. El mussol pot recol·lectar elements extra que li donen característiques especials. Per exemple invencibilitat o que els propers elements que caiguin siguin només estrelles.

I petits detalls que són millorables com,

- La posició dels botons que canvien de color. De vegades apareixen una mica tallats per la part inferior.
- Adaptar l'aplicació a mode *landscape*.

Ara bé, hi ha dos aspectes externs a l'aplicació que caldria millor buscant altres alternatives a html, css, javascript i P5.js. Aquests són la renderització de l'aplicació i el temps de càrrega dels arxius.

Capítol 5: Demostració

1. Instruccions d'instal·lació

El producte final està en format *Web App*, això vol dir que només cal un mòbil i un navegador per fer-lo funcionar.



Figura 17: QR Super Owillie Web App

L'aplicació està allotjada a 000webhost.com una pàgina que ofereix un servei de *hosting* gratuït. Es pot accedir al producte a través del següent *link*: <https://superowillie.000webhostapp.com/> o si es prefereix també s'hi pot accedir a partir del codi QR.

Preparació per visualitzar l'aplicació com a *Web App* (mode pantalla completa):

1. Obrir el navegador del mòbil i escriure a la barra de direccions l'enllaç proporcionat. També es pot obrir l'enllaç a partir del codi QR proporcionat.
2. Crear l'accés directe.
 - En iOS s'ha de prémer el botó "compartir" (quadrat amb fletxa) del navegador i seleccionar l'opció "afegir a pantalla d'inici".
 - En Android s'ha de prémer el botó "menú" (tres puntets verticals) del navegador i seleccionar l'opció "afegir a pantalla d'inici".
3. Un cop creat l'accés directe es crearà, al que seria l'escriptori del *smartphone*, una icona d'un mussol observant la lluna. Ja es pot executar l'aplicació prement aquesta icona.

Consideracions de hardware

Al tractar-se d'una *Web App* aquesta aplicació es pot executar en diferents tipus de dispositius tant en mòbils com *tablets* com ordinadors. Tot i així aquest projecte només s'ha centrat en mòbils i per això es recomana fer ús d'un *smartphone* per executar l'aplicació.

Si s'usa un dispositiu de major resolució de pantalla (*tablet/pc*) pot passar que alguns dels botons no siguin del tot clicables o que alguna funcionalitat no funcioni correctament. L' aspecte visual del joc també s'hauria d'adaptar millor o, fins i tot, fer 5 columnes de baixada d'elements en comptes de 3.

Consideracions de sistema

- Les proves s'han realitzat principalment en dispositius amb sistema iOS. S'han usat dispositius de diferent resolució de pantalla: iPhone 5s, iPhone 6, iPad Air i iPod 5ª generació. En tots aquests dispositius el seu funcionament és correcte.
- En Android també s'han realitzat algunes proves tot emprant mòbils de diferents games i fabricants. El resultat ha estat que depenent de les característiques i potència de cada mòbil, el funcionament del joc varia. En alguns casos el joc funcionava correctament però ralentitzat.
 - Mòbils amb bon funcionament: Note 2 de Samsung, Xiaomi MI5 i Huawei p9.

2. Mapatge de l'aplicació

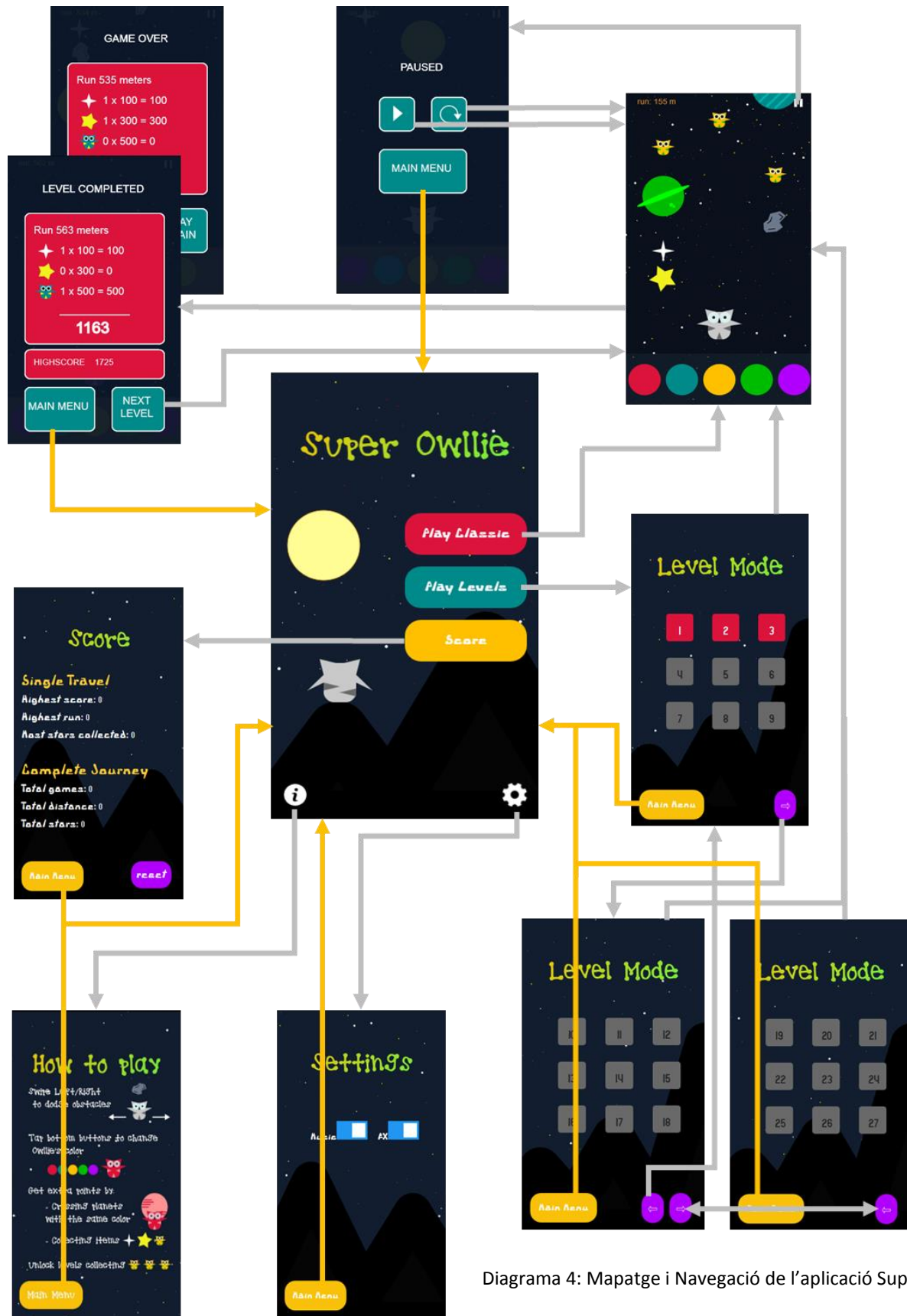


Diagrama 4: Mapatge i Navegació de l'aplicació Super Owillie

3. Tests

3.1 Test usuari: aplicació

L'etapa de disseny conclouia amb l'elaboració d'un prototip. Aquest prototip se centrava exclusivament en l'arquitectura i navegació de l'aplicació descrites en l'etapa de disseny.

L'aplicació presenta una estructura bàsica i senzilla, la seva funcionalitat és permetre un accés ràpid al joc i, com a molt, afegir alguna funcionalitat extra com són les puntuacions. En general es tracta d'una navegació en dos nivells (pantalla inicial + pantalla secundària) menys en la pantalla *Levels* que és de quatre nivells. No es tracta d'una aplicació complexa i per això no s'espera que es detectin problemes greus durant les proves d'usuari.

Les proves d'usuari han servit per confirmar l'estructura de l'aplicació i comprovar si es complien certs objectius. Aquestes han consistit en demanar a 10 persones que realitzessin una sèrie de tasques:

- **Tasca 1: Fes una partida.**
- **Tasca 2: Reinicia la partida.**
- **Tasca 3: Carrega el nivell 5.**
- **Tasca 4: Mou-te lliurement per l'aplicació.**

A partir d'aquí s'ha volgut saber si podien diferenciar entre els dos modes de joc i si en algun moment no han sabut com tornar a la pantalla d'inici.

Els resultats han estat els esperats. La única consideració a tenir en compte és la pantalla intermèdia entre el menú principal i el joc en mode *Classic*. Al arribar a aquesta pantalla molts s'han preguntat perquè s'havia de seleccionar la dificultat si ja existeix un mode *Level*.

Arran d'aquesta troballa s'ha decidit suprimir aquesta pantalla i, en el seu defecte, aquesta funcionalitat extra del joc. Ara el mode clàssic tindrà per defecte sempre cinc possibles canvis de color (cinc botons).

3.2 Test usuari: joc

Finalitzada l'etapa d'implementació i amb el joc gairebé acabat (a espera de lidiar amb els últims detalls) s'ha volgut provar el seu funcionament, així com també comprovar la resposta del públic entorn al joc.

Aquest test d'usuari ha consistit en demanar a 20 persones que juguessin lliurement amb el joc. Se'ls hi ha proporcionat la pàgina web on està allotjat el joc i se'ls hi ha donat instruccions per a que l'instal·lessin a la pantalla d'inici.

Ha tingut una acceptació molt bona. En general ha resultat un joc divertit i que ha "enganxat". Tot i així s'han detectat una sèrie de *bugs* que s'han intentat solucionar (alguns han estat impossibles de solventar) i s'ha vist que en alguns casos ha costat entendre què s'havia de fer.

El joc en sí és fàcil d'entendre i d'interactuar-hi. La única cosa poc intuïtiva és entendre que el mussol col·lideix també amb els planetes però que aquests poden ser travessats si el mussol exhibeix el mateix color. Per tal de solucionar aquest aspecte, s'ha procedit a afegir una nova pantalla que és la pantalla *Help*. En aquesta s'explica quins són els moviments a realitzar, s'ensenya que els botons de la part inferior fan canviar el color del mussol i s'indica que el mussol no col·lideix amb els planetes del mateix color. D'aquesta manera queda ara una mica més clar què s'ha de fer.

Un altre aspecte a tenir en compte és que per a realitzar la prova s'han utilitzat mòbils diversos. En alguns casos, aquests mòbils no eren suficientment potents per a reproduir el joc i el ralentitzaven. Sobretot en mòbils Android. D'altra banda en iPhone no hi ha hagut cap problema, sobretot l'iPhone 6 i 6s on ha rendit a la perfecció.

Capítol 6: Conclusions i línies de futur

1. Conclusions

Una de les frases més captivadores d'aquests estudis de màster és que els llenguatges web permeten construir una única aplicació que funciona en tots els dispositius mòbils. La intenció d'aquest projecte final ha estat precisament constatar que aquesta afirmació es compleix. Per això s'ha dissenyat i implementat una aplicació emprant només llenguatges web.

El resultat ha estat sorprenent. Més extraordinari del què em pensava!

L'aplicació s'executa en Android i iOS, en mòbil, *tablet* i ordinador i en tots els navegadors. Uns performen millor que d'altres però és fascinant com escrivint el codi un sol cop, aquest funciona en els diferents dispositius. Tot i així sí que és veritat que quan un implementa una aplicació ha de tenir present que existeixen diversos sistemes operatius i que no tot funciona igual. De vegades cal implementar alguna cosa de diferent manera. Seguint aquesta mateixa línia també he pogut comprovar que el sistema d'Android és més permissiu que el d'Apple.

A nivell de projecte, considero que tot i haver hagut d'allargar la planificació aquesta era força correcta. Si ara tornés a realitzar aquest projecte escurçaria l'etapa de disseny i allargaria l'etapa d'implementació. No tant per què consideri que he previst poc temps a la implementació, però sí perquè en aquesta etapa sorgeixen més imprevistos que en la de disseny. I donat que l'experiència és un factor clau a l'hora de solucionar els imprevistos, jo he necessitat encara més temps per fer-ho.

De totes maneres el procés de treball emprat, la distribució del projecte en tres fases i les tasques definides m'han permès seguir bé la planificació i tirar endavant el projecte. En aquest aspecte no s'han hagut d'afegir canvis al projecte.

Els objectius s'han complert, s'ha implementat una *game app* amb un joc del gènere *endless running*. La caracterització dels personatges i elements de joc responen a allò que es volia aconseguir. En definitiva, es volia crear un joc divertit i que agradés. Per això l'alegria més gran va ser durant les proves d'usuari del joc quan vaig veure la gran acceptació que el joc tenia entre els usuaris.

A nivell global, estic més que satisfeta del resultat final d'aquest projecte. Han estat uns mesos intensos però que m'han fet passar una bona estona, entretinguda i divertida. Especialment en aquells moments en els que les coses no sortien i he hagut de replantejar el problema i buscar-hi alternatives. Ha estat un projecte que m'ha permès posar en pràctica tot l'aprens durant el màster però que també ha tingut una part important d'autoaprenentatge.

En general he gaudit de cadascuna de les diferents fases del projecte, però la part d'implementació és la que més m'ha apassionat. El fet de crear qualsevol cosa a partir de codi, les diverses maneres d'abordar una mateixa funcionalitat, la interconnexió amb els diferents llenguatges...

Estic orgullosa de veure com el que era una simple idea s'ha acabat convertint en un producte real.

2. Línies de futur

Des de l'inici he tractat aquest projecte/producte com una primera versió. Amb això vull dir que no és un producte tancat al que ja no se li poden fer canvis. Tot el contrari. Aquest és un producte que encara permet l'addició de noves funcionalitats.

A banda de la inclusió de noves funcionalitats, el que sí seria important és:

- Solucionar els *bugs* i implementar totes aquelles funcions que no han donat temps a realitzar.
- Adaptar l'aplicació per a altres dispositius com *tablets* i ordinadors. No es tracta simplement de fer que funcioni en aquests dispositius sinó aprofitar que la pantalla és de major tamany i adaptar el concepte de joc. Una opció seria que en comptes de tenir tres posicions de moviment del mussol, fer-ne cinc. Així s'aprofitaria millor la pantalla.
- Adaptar l'aplicació per ambdues orientacions o, si més no, bloquejar el canvi d'orientació a horitzontal.
- Millorar la renderització i la rapidesa de càrrega. En aquest projecte no s'ha emprat cap motor de joc. No hi ha hagut temps per aprendre una nova tecnologia com Coco2 o Phaser, però aquestes serien dos possibles opcions a considerar.
- Aprofitar les característiques que ofereixen avui en dia els *smartphones* i fer, per exemple, que el mussol es mogui a través de l'acceleròmetre del mòbil.
- Adquirir els drets de les músiques i fonts i distribuir l'aplicació per les principals botigues online: Google Play i App Store.

Bibliografia

Llibres - Manuals

Arce Anguiano, F. J. (2014). *Programa Juegos con HTML5*. Mexico: Alfaomega

Geary, D. (2014). *Core HTML5 Canvas*. Indiana, US: Prentice Hall.

VV.AA. (2012). *Desarrollo de juegos en HTML5*. Madrid, Espanya: Anaya.

Articles

“7 Best HTML5 Based Frameworks For Developing Cross-Platform Apps” (2015). Cohen, Addison.

<http://codecondo.com/7-best-html5-based-frameworks-for-developing-cross-platform-apps>
[consultat: 24/10/16]

“Planificación y gestión de proyectos de desarrollo de videojuegos “ (2015). Ferrán, Elío.

<http://es.slideshare.net/elioferran/gestion-de-proyectos-videojuegos>
[consultat: 28/10/16]

“Angry Birds: the story behind iPhone's gaming phenomenon” (2011). Kendall, Paul.

<http://www.telegraph.co.uk/technology/video-games/8303173/Angry-Birds-the-story-behind-iPhones-gaming-phenomenon.html>
[consultat: 17/11/16]

“How to Add Icons for iPhone, iPad & Android to Your Website” (2017). Larson, Kyle.

<http://www.kylejlarson.com/blog/adding-an-icon-for-iphone-ipad-android-to-your-website>
[darrera consulta: 10/01/17]

“Which is better for mobile app development, Intel XDK or Sencha?” (2015). Limbasiya, Rakesh.

<https://www.quora.com/Which-is-better-for-mobile-app-development-Intel-XDK-or-Sencha>
[consultat: 26/10/16]

“El sector del videojuego atraviesa una segunda época dorada” (2015). Paniagua, Rebeca.

<http://www.elmundo.es/tecnologia/2015/07/19/55ab899322601d9f518b456f.html>
[consultat: 19/10/16]

“Construye apps móviles usando HTML5 con Intel XDK” (2013)

<https://sg.com.mx/buzz/construye-apps-moviles-usando-html5-intel-xdk#.WA6X3YOLTIU>
[consultat: 26/10/16]

“Temple Run 2 interview” (2013)

<https://www.telegraph.co.uk/technology/video-games/9811529/Temple-Run-2-interview.html>

[consultat: 20/10/16]

Web

Intel® XDK website: <https://software.intel.com/es-es/intel-xdk>

[consultat: 28/12/16]

P5.js website: <https://p5js.org>

[darrera consulta: 14/01/17]

Stackoverflow website: <http://stackoverflow.com/questions>

[darrera consulta: 14/01/17]

W3schools website: <http://www.w3schools.com>

[darrera consulta: 13/01/17]

Wikipedia website: https://en.wikipedia.org/wiki/Platform_game#Endless_running_game
“Platform Game”

[consultat: 19/10/16]

Wikipedia website: https://en.wikipedia.org/wiki/Game_design_document “Game design document”

[consultat: 28/10/16]

Video

Shiffman, D. [Daniel Shiffman]. (2016, octubre 16). 7.4: Handling DOM Events with Callbacks

- p5.js Tutorial [Arxiu de video]. Recuperat de

<https://www.youtube.com/watch?v=NcCEzdz9BGE&t=147s>

Shiffman, D. [Daniel Shiffman]. (2016, maig 20). 7.5: Interacting with the DOM using Sliders,

Buttons and Text Inputs - p5.js Tutorial [Arxiu de video]. Recuperat de

<https://www.youtube.com/watch?v=587qclhguQg&t=555s>

Shiffman, D. [Daniel Shiffman]. (2016, octubre 16). 7.7: The Basics of CSS - p5.js Tutorial

[Arxiu de video]. Recuperat de <https://www.youtube.com/watch?v=zGL8q8iQSQw&t=506s>

Shiffman, D. [Daniel Shiffman]. (2016, juny 7). 11.1: Loading and Playing - p5.js Sound

Tutorial [Arxiu de video]. Recuperat de https://www.youtube.com/watch?v=Pn1g1wjxl_0

Shiffman, D. [Daniel Shiffman]. (2016, juny 8). 11.2: Play and Pause Button - p5.js Sound Tutorial - p5.js Sound Tutorial [Arxiu de video]. Recuperat de <https://www.youtube.com/watch?v=YcezEwOXun4>

Imatges

"PacMan Flow Chart" [img]. Fugiel, Richie. <http://rfugiel.com/images/pacman.png>
[consultat 5/11/16]

"Flowchart game super mario bross" [img] (2014). Makrus, Ali. http://1.bp.blogspot.com/-Z4y0HFo8Shc/VArTLiZ_yCI/AAAAAAAAACA/qHf9Lkm3bOw/s1600/mario-input-basics.jpg
[consultat 5/11/16]

Annexos

Annex A: Glossari

App Store: servei pels productes d'Apple que permet als usuaris descarregar aplicacions.

Arcade: gènere de videojocs dels anys 80 que es caracteritzen per tenir una mecànica molt senzilla.

Bug: error o defecte en el software que fa que el programa faci coses que nno hauria de fer.

Canvas: és un element d'HTML5 que permet dibuixar gràfics, manipular imatges i realitzar animacions en una pàgina web de forma dinàmica.

CSS: Cascade Style Sheet.

Endless running: gènere de videojoc que consisteix en que el personatge avança indeterminadament fins que impacta amb algun objecte o bé cau en algun forat.

Game design document: document de disseny de videojocs. És un document descriptiu de conceptualització de joc.

Game play: conjunt d'accions que realitza un jugador per interactuar amb el joc.

Gamer: persona que juga sovint als videojocs. Se'l caracteritza per dedicar-se amb gran passió i interès als jocs en general.

Google Play: servei pels productes de Google/Android que permet als usuaris descarregar aplicacions.

HTML: HyperText Markup Language traduït com llenguatge de marques d'hipertext.

Javascript: és un llenguatge de programació que es pot utilitzar per construir llocs web i per fer-los més interactius.

P5.js: una llibreria de javascript per a la programació visual que segueix les bases de Processing.

PAC: Prova d'Avaluació Continuada a la UOC.

Power ups: dins d'un videojoc són potenciadors o elements especials que donen poders.

Swipe: moviment tàctil de que es realitza lliscant un dit per la pantalla.

Tap: moviment tàctil que equival a clicar qualsevol botó interactiu.

TFM: Treball Final de Màster.

Wireframing: és una manera de dissenyar una aplicació des d'un nivell estructural. També conegut com esquema de pàgina o pla de pantalla.

Annex B: Lliurables del projecte

En aquesta entrega es presenten els següents lliurables:

- Informe de seguiment de la PAC 5.
- Memòria del projecte.
- Codi Font de l'aplicació i el joc.
- Presentació acadèmica en format vídeo
- Presentació pública en format power point