

Desenvolupament d'un joc tipus "Shoot'em up" per Android amb opció multi-jugador. "Hero Shooter"

Oscar Ferrer Travesset

Pla d'estudis: Enginyeria Tècnica de Telecomunicació, especialitat en Telemàtica

Consultor: Manel Llopart Vidal

8 de Gener de 2014



Aquesta obra està subjecta a una llicència de

[Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

És lliure de:

- **Compartir** — copiar i redistribuir el material en qualsevol mitjà o format.
- **Adaptar** — remesclar, transformar i crear a partir del material.
- El llicenciant no pot revocar aquestes llibertats mentre compleixi els termes de la llicència.

Sempre que es compleixin les següents condicions:

- **Reconeixement** — S'ha de reconèixer adequadament la autoria, proporcionar un enllaç a la llicència i indicar si s'han realitzat canvis. Es pot fer de qualsevol manera raonable, però no de una manera que suggereixi que disposa del recolzament del llicenciant o que el rep per l'ús que en fa.
- **No Comercial** — No es pot utilitzar el material per a una finalitat comercial.
- **Compartir Igual** — Si remescla, transforma o crea a partir del material, es tindrà que difondre les contribucions sota la mateixa llicència que l'original.
- **No restriccions addicionals** — No es poden aplicar termes legals o mesures tecnològiques que legalment restringeixin realitzar el que la llicència permet.

Fitxa del Treball Final

Títol del treball:	Desenvolupament d'un joc tipus "Shoot'em up" per Android amb opció multi-jugador. "Hero Shooter"
Nom de l'autor:	Oscar Ferrer Travesset
Nom del consultor:	Manel Llopart Vidal
Data de lliurament:	01/2014
Àrea del Treball Final:	Aplicacions multimèdia de nova generació
Titulació:	Enginyeria Tècnica de Telecomunicació, especialitat en Telemàtica

Paraules clau

- Android
- Joc
- Multi-jugador
- Shoot'em Up
- Google Play Game Services
- Canvas
- Android SDK

Abstract

Aquest projecte es basa en el desenvolupament d'un joc per Android amb mode multi-jugador i funcionalitats socials. El joc és del tipus "Shoot'Em Up", on es tracta de controlar un personatge/heroi, que avança de forma indefinida, mentre apareixen enemics i obstacles que cal esquivar o disparar. L'objectiu del joc és recórrer el màxim de metres sense morir, aconseguint el màxim de puntuació possible.

Val a dir, que el joc disposa de capacitat per connectar-se amb la xarxa social de Google, i utilitzar el compte de l'usuari per registrar les diferents fites que aquest va assolint en el joc i comparar les puntuacions obtingudes entre les seves amistats de Google+ o inclús la resta d'usuaris de la xarxa social. Mitjançant els serveis de Google, el joc permet realitzar partides multi-jugador online, realitzant enfrontaments amb usuaris aleatoris o bé desafiant amics de la xarxa social.

Així, per poder entendre el context en el que es situa el projecte, en aquest treball es realitza un breu anàlisi històric sobre l'aparició dels Smartphones i les utilitats que aquests han adquirit.

Per altra banda, es resumeixen les característiques principals dels sistemes operatius mòbils més comuns, centrant l'atenció especialment sobre Android. Respecte aquest, a més se'n descriu el sistema i funcionament a nivell de programació, encaminant les explicacions al desenvolupament d'un joc.

En aquest treball s'analitza doncs, el funcionament del motor gràfic i renderitzat d'un joc en Android per tal d'entendre'n la estructura interna. Així, s'exposa el funcionament del bucle del joc desenvolupat, tenint en compte aspectes com la velocitat del joc, els FPS (Frames Per Segon) i el renderitzat de les imatges.

Per altra banda, s'exposa breument el funcionament de les llibreries de Google Play Games Services, que han servit per desenvolupar les funcionalitats socials i l'apartat multi-jugador del joc.

Este proyecto se basa en el desarrollo de un juego para Android con modo multijugador y funcionalidades sociales. El juego es del tipo “Shoot’Em Up”, donde se trata de controlar un personaje/héroe, que avanza de forma indefinida, mientras aparecen enemigos i obstáculos a esquivar o disparar. El objetivo del juego es recorrer el máximo de metros sin morir, consiguiendo el máximo de puntuación posible.

Hay que destacar, que el juego dispone la capacidad de conectarse con la red social de Google, y utilizar la cuenta del usuario para registrar los logros que vaya realizando en el juego y comparar las puntuaciones obtenidas con sus amistades de la red social, o incluso con la resta de usuarios de Google+. Mediante los servicios que ofrece Google, el juego permite jugar partidas multi-jugador en línea, realizando enfrentamientos contra usuarios aleatorios o desafiando amigos de la red social.

Para comprender el contexto en el que se sitúa el proyecto, se realiza un breve análisis histórico sobre la aparición de los Smartphones y las utilidades que estos han adquirido.

Por otro lado, acotando el contexto sobre el que se desarrolla el proyecto, se resumen las características principales de los sistemas operativos móviles más comunes, centrando la atención especialmente sobre Android. Respecto a éste, además se describe el sistema y funcionamiento a nivel de programación, encarando las explicaciones hacia el desarrollo de un juego.

En este trabajo, se analiza el funcionamiento del motor gráfico y renderizado de un juego en Android para poder así entender su estructura interna. Posteriormente, se expone el funcionamiento del bucle de juego desarrollado, teniendo en cuenta aspectos como la velocidad del juego, los FPS (Frames Por Segundo) y el renderizado de las imágenes.

Por otro lado, se expone brevemente el funcionamiento de las librerías de Google Play Games Services, que han servido para desarrollar las funcionalidades sociales y el apartado multijugador del juego.

This project is based on developing an Android game with multiplayer mode and social features. The game is a "Shoot'Em Up", where you must control a character / hero, which progresses indefinitely while enemies and obstacles appear, having to shoot or dodge them. The object is to cover the maximum meters without dying, acquiring the maximum score as possible.

Of note, the game features the ability to connect with the social network of Google, and the user account used to record the achievements that you perform in the game and compare scores with friends of social network, or even with the rest of Google+ users. Through the services Google provides, the game allows you to play multi-player games online, performing matches against random users or defying social network friends.

To understand the context in which the project takes place, first it's made a brief historical analysis of the emergence of Smartphones and their capabilities.

On the other hand, limiting the context on which the project is developed, it's summarized the main features of the most popular mobile operating systems, focusing especially on Android. Regarding this, in addition is described the programming system, facing the explanations on developing a game for Android.

In this paper, the operation about graphic and rendering engine of a game in Android is analyzed in order to understand its internal structure. Subsequently, the operation of the game loop developed is exposed, taking in mind aspects such as the speed of the game, the FPS (Frames Per Second) and rendering of images.

On the other hand, describes briefly the operation of libraries Google Play Games Services, which served to develop the social features and multiplayer section of the game.

Resum

En aquest projecte es pretén realitzar un joc del tipus “Shoot’em up” amb opció de partides multi-jugador per a dispositius Android. Aquest gènere de videojocs es defineix pel fet que l’usuari controla un únic personatge que ha de disparar contra els enemics que van apareixent en pantalla. Generalment, el fons de pantalla del joc, avança de dreta a esquerra durant tota la partida, d’aquesta manera, el personatge que es controla avança d’esquerra a dreta, sense que s’aturi el moviment en cap moment (representa que el personatge avança indefinidament en direcció horitzontal), mentre van apareixent enemics en direcció oposada, que l’usuari ha de disparar o esquivar.

Es tracta d’un joc en dues dimensions, pensat per realitzar partides relativament curtes, on l’objectiu de cada partida és avançar el màxim de metres possible evitant qualsevol col·lisió del personatge amb els obstacles enemics. Inicialment, es disposa d’uns moviments molt limitats i a mesura que es realitzen partides s’obtenen recursos, de manera que l’usuari pot “comprar” nous moviments per al personatge. Així, en un principi només es pot córrer, saltar i disparar per evitar els enemics. Posteriorment, a base de compres es pot adquirir l’habilitat de saltar múltiples vegades a l’aire o fins i tot volar.

A mesura que s’avança en una partida, s’incrementa un comptador de metres situat a la part superior de la pantalla, d’aquesta manera, la puntuació de cada partida serà el nombre de metres avançats abans de morir. Cada vegada que es mata un enemic, aquest deixa al seu lloc una moneda i es suma 100 punts al total de la partida, d’aquesta manera s’incentiva el joc, no només incitant a esquivar enemics sinó animant a disparar-los i arriscar per obtenir millors puntuacions. Així, es disposa d’un apartat de puntuacions en mode local i en mode compartit, on mitjançant el compte de “Google +” es podrà comparar les puntuacions obtingudes per l’usuari amb la resta dels seus contactes de la xarxa social que hagin jugat a aquest joc o amb les puntuacions globals de tots els jugadors.

Per altra banda, el joc disposa de mode multi-jugador, on el joc connecta amb un servidor extern per tal de trobar altres jugadors i poder desafiar-los. En trobar un oponent, cada jugador iniciarà una partida normal de forma individual. El guanyador resultarà el que hagi obtingut una puntuació superior durant la seva partida particular.

Agraïments

A la Maite,

per tot el suport que m'ha donat, per com m'ha cuidat i animat durant tot el projecte i per aguantar-me en els millors i pitjors moments.

A la Mama,

pels ànims i tot el suport que m'ha donat, sempre amb un plat preparat per quan no hi ha temps per cuinar.

Al Marc,

pels ànims que m'ha donat, creient en mi des del primer moment.

Al Manel,

per tots els comentaris en cada PAC i per com m'ha guiat en el projecte, des de escollir el tema fins a la realització d'aquesta memòria.

Índex

1.	Introducció	12
1.1	Context i justificació del Treball	12
1.1.1	Motivació personal.....	12
1.1.2	Beneficis	13
1.2	Objectius del Treball.....	14
1.3	Enfocament i mètode seguit	15
1.3.1	Recursos inicials	18
1.4	Planificació del Treball	21
1.4.1	Diagrama de Gantt	22
1.4.2	Desviació	24
1.5	Breu sumari de productes obtinguts.....	25
2.	Estat de l'art	28
2.1	Telèfons mòbils intel·ligents: l'aparició dels "Smartphones"	28
2.1.1	Usos freqüents	30
2.1.2	Capacitats per jugar.....	31
2.2	Sistemes Operatius Mòbils.....	32
2.2.1	iOS	32
2.2.2	Windows Phone	33
2.2.3	BlackBerry.....	33
2.2.4	Android.....	35
2.3	Jocs en Smartphones.....	39
2.3.1	Tipus de jocs més utilitzats.....	40
2.3.2	Exemples de jocs	41
2.4	Programar per Android	43
2.4.1	Sistema de desenvolupament	44
2.4.2	Desenvolupament de jocs: Frameworks gràfics.....	45
3.	Disseny i implementació	48
3.1	Estructura del joc	48
3.1.1	Base del joc.....	48
3.2	Funcions extra: Google +.....	51
3.2.1	Descripció Google Play Games Services (GPGS).....	51
3.2.2	Aplicació de GPGS al joc desenvolupat	51
3.3	Multi-jugador	53
3.3.1	Inici de la partida multi-jugador	59
3.4	Apartat gràfic: Renderitzat.....	62

4.	Entorn de programació	69
4.1	Configuració Eclipse	69
4.2	Google Play Games Services.....	71
4.3	Fons de codi re-utilitzat	72
5.	Conclusions	74
5.1	Línies de treball futur	74
6.	Glossari.....	76
7.	Bibliografia	80

Índex de figures

Figura 1. Detall de la planificació temporal.....	21
Figura 2. Diagrama de Gantt – PACs 1 i 2.....	22
Figura 3. Diagrama de Gantt – PAC 3.....	22
Figura 4. Diagrama de Gantt – Detall fase de disseny.	22
Figura 5. Diagrama de Gantt – Detall fase d’implementació.	23
Figura 6. Diagrama de Gantt – Documentació.....	24
Figura 7. Diagrama de Gantt – PACs 4 i 5.....	24
Figura 8. Captures de pantalla – Pantalla inicial.	26
Figura 9. Captures de pantalla – Fites i puntuacions.	26
Figura 10. Captures de pantalla – Botiga de bons i multi-jugador.....	27
Figura 11. Captures de pantalla - Partida.....	27
Figura 12. Taula de versions Android i percentatge d’ús. [6].....	37
Figura 13. Gràfic de versions Android i percentatge d’ús. [6].....	38
Figura 14. Imatges de Jocs – Candy Crush Saga.....	41
Figura 15. Imatges de Jocs – Ralph’s World.....	42
Figura 16. Imatges de Jocs – Jetpack Joyride.....	42
Figura 17. Imatges de Jocs – Angry Birds.....	43
Figura 18. Diagrama de Flux – Partida simple.....	49
Figura 19. Diagrama de blocs – Base de dades de puntuacions.	50
Figura 20. Diagrama de Flux – inici/fi sessió Google +.....	52
Figura 21. Diagrama de Flux – Mode multi-jugador.	58
Figura 22. Diagrama de Flux – Inici partida multi-jugador.....	61
Figura 23. Renderitzat – 1FPS. [7].....	64
Figura 24. Renderitzat – 10 FPS. [7].....	64
Figura 25. Renderitzat – Situació 2. [7].....	65
Figura 26. Renderitzat – Situació 3. [7].....	66
Figura 27. Renderitzat – Situació real. [7].....	67
Figura 28. Gràfics del joc – Monedes.....	68
Figura 29. Gràfics del joc – Heroi.....	68
Figura 30. Configuració de l’IDE – SDK Manager.....	70
Figura 31. Configuració de l’IDE – Ubicació SDK.....	71

1. Introducció

1.1 Context i justificació del Treball

L'aparició dels Smartphones ha suposat una revolució en el món de la telefonia mòbil, però no només en l'àmbit tecnològic sinó també en el social. Els telèfons intel·ligents han canviat la manera en que les persones utilitzen els mòbils, aportant certa quotidianitat en el seu ús, fins a convertir-se en indispensables per al dia a dia de molta gent. Aquest fet pot arribar a ser discutible a nivell moral, però el que no es pot negar és els grans avantatges que aquests aporten pel que fa a connectivitat i disponibilitat de la informació.

Actualment es viu en la era de la informació, i els Smartphones juguen un rol molt important en aquesta revolució, permetent a tothom estar connectat en tot moment amb la resta del món via internet. Gràcies a aquests dispositius de butxaca (alguns ja no tant), les persones tenen a l'abast de la seva mà un món de possibilitats que anys enrere era inimaginable, aportant noves funcionalitats en àmbits que no es plantejaven per a dispositius mòbils, com per exemple l'entreteniment i l'oci a través dels jocs i l'aparició de les xarxes socials.

Així, el percentatge de Smartphones respecte a telèfons mòbils tradicionals s'ha vist incrementat de forma significativa en els últims anys. Actualment a Europa el 57% de dispositius mòbils ja són Smartphones, dels quals un 70% utilitzen el sistema operatiu Android. Aquestes xifres augmenten dia rere dia i si bé segueix sent cert el que es deia anys enrere respecte a que el futur passa pels dispositius mòbils intel·ligents i les aplicacions mòbils, cal tenir molt clar que aquests, avui en dia ja formen part del present, i per tant representen un pou d'oportunitats per poder arribar a un nombre cada vegada major d'usuaris.

1.1.1 Motivació personal

Personalment vaig descobrir el món de la programació per Android de forma totalment autodidacta com a resultat d'una curiositat que feia temps que tenia. A principis de 2013, just després de finalitzar els exàmens del primer semestre del curs i durant el buit de temps entre final de semestre i l'inici del següent, vaig decidir informar-me pel meu compte propi sobre el funcionament d'Android a nivell de programació. Aleshores vaig entrar a un nou món desconegut per mi fins aleshores, que em va fascinar tant que des de llavors no he pogut parar. Així, poc a poc he anat aprenent els fonaments de la programació per Android a base de blogs, vídeo guies i la documentació oficial de Google.

Posteriorment, em vaig inscriure a un curs de programació per Android a la Universitat Politècnica de València que s'impartia durant l'estiu. D'aquesta manera, vaig poder consolidar els coneixements adquirits de forma autodidacta fins aleshores.

Una de les raons sobre l'interès personal per Android, és el seguiment que he tingut sempre envers el sistema operatiu des dels seus inicis, quan encara poca gent el coneixia. A mesura que ha anat avançant el temps, el sistema ha millorat considerablement, i les aplicacions per aquest també ho han fet de forma simultània, però sempre he trobat objeccions o possibles defectes a moltes aplicacions o directament funcionalitats inexistentes, i aquest fet m'ha portat a endinsar-me en la programació per Android.

Per altra banda, sempre m'ha cridat la atenció el desenvolupament de jocs, tot i que mai abans m'havia atrevit a endinsar-m'hi a nivell de programació, i donat que en el curs de la UPV es toca el tema del desenvolupament de jocs, encara que sigui a un nivell molt superficial, ja m'ha aportat una petita base conceptual sobre aquest tipus de desenvolupament. Així, em va agradar molt la idea de poder cursar el Treball de Final de Carrera relacionat amb la programació d'un joc per Android.

Pel que fa a la idea del joc, he escollit un "Shoot'em Up", ja que es un tipus de joc que personalment trobo molt engrescador i entretingut, amb partides relativament curtes, però amb al·licients suficients per enganxar l'usuari de forma duradora, ja sigui a base de nivells, millores d'armes o d'habilitats. De totes maneres, tot i que hi ha multitud de jocs d'aquest estil ja creats i publicats al "Play Store", per al meu gust a tots els hi falta alguna cosa, el que té un li falta a l'altre i viceversa. Per aquesta raó, he decidit engegar aquest projecte amb aquesta idea, crear el "Shoot'em Up" ideal, si més no per als meus gustos, i a més donar-li la possibilitat de partides multi-jugador, funcionalitat inèdita en aquest gènere.

1.1.2 Beneficis

Els beneficis d'aquest projecte estaran relacionats bàsicament amb l'entreteniment. Al tractar-se d'un joc, el que pretén aportar és entreteniment i diversió cap als seus usuaris, però principalment de la forma més duradora possible.

Al tractar-se d'un joc de partides relativament curtes, permetrà als seus usuaris jugar en qualsevol lloc, a la parada de l'autobús o en qualsevol moment d'espera que es pugui trobar, és a dir, una manera d'emplenar els petits buits de temps mort que es donen en la societat d'avui en dia.

Per altra banda, donat que els dispositius mòbils en una gran majoria, ja disposen de connexió a internet, el joc aprofita aquesta connectivitat permetent partides en mode multi-jugador, afavorint la mobilitat i el rang d'espais o moments on poder-hi jugar.

Un tret important del joc, es que està pensat per tenir una dificultat incremental de forma molt gradual, permetent millorar les opcions de joc paulatinament en aconseguir realitzar uns reptes proposats. D'aquesta manera, l'usuari sempre trobarà un al·licient per seguir jugant, evitant la sensació d'avorriment o cansament envers la repetició del joc.

1.2 Objectius del Treball

Els objectius d'aquest projecte són principalment desenvolupar un joc per Android amb capacitats per realitzar partides multi-jugador, de manera que durant el procés de disseny i implementació, permeti aplicar els conceptes adquirits durant l'enginyeria tècnica de telecomunicacions en el màxim d'assignatures possible.

Per altra banda, es pretén ampliar els coneixements propis pel que fa al desenvolupament d'aplicacions per Android, entrant en el món de la programació de videojocs, aprenent el funcionament de llibreries externes com LibGDX o AndEngine per l'apartat de gràfics i física del joc, o com Google Play Game Services o Google App Engine per l'apartat multi-jugador.

Pel que fa als conceptes adquirits durant la carrera, es pretén posar en pràctica sobretot el que s'ha après en "Fonaments de Programació", "Programació Orientada a Objectes" i "Sistemes Operatius" ja que el desenvolupament del joc serà principalment una aplicació directa d'aquestes dues assignatures, més tenint en compte que el llenguatge de programació utilitzat per programar per Android és el Java, i la lògica estructural que s'utilitza es basa en l'orientació a objectes. Per altra banda, serà necessari tenir en compte el comportament i el cicle de vida de les aplicacions i processos del Sistema Operatiu en qüestió.

En segon lloc, però no menys important, a banda de la programació en sí, es pretén aplicar els diversos coneixements adquirits pel que fa al camp de les matemàtiques, la física i la estadística, ja sigui pel control del personatge, el moviment i la física del joc, com per als esdeveniments aleatoris que es produiran en una partida, que prèviament s'han de preveure i programar.

Per altra banda, l'apartat visual del joc i la interacció de l'usuari hauran de ser un punt important per al resultat final, ja que és el primer que es veu i per tant és la primera impressió que s'enduran els usuaris, així, s'haurà de tenir en compte tot el que s'ha estudiat en la optativa "Interacció Humana amb els Ordinadors".

Finalment, com que es tracta d'un joc amb possibilitat de jugar partides multi-jugador, serà imprescindible fer ús dels coneixements adquirits en les assignatures de "Protocols i Aplicacions d'Internet" i "Administració de Xarxes i Sistemes Operatius" pel que fa a la connexió entre els usuaris i un servidor extern per gestionar les partides multi-jugador, mitjançant l'ús dels Sockets o la gestió de les connexions mitjançant llibreries externes.

Sense oblidar però, que durant la elaboració de tot el projecte s'haurà de seguir una organització i planificació del temps ben definida i per tant, s'haurà d'aplicar els mètodes assolits amb l'assignatura de "Gestió de Projectes", de manera que finalment es pugui complir amb els objectius aquí esmentats, entregant com a resultat final la memòria, el joc instal·lable juntament amb el codi font i una presentació del projecte.

Per tant, es pot resumir l'anterior en:

- Desenvolupar un joc per Android amb opció de partides multi-jugador.
- Ampliar els coneixements en el desenvolupament per Android.
- Aprendre sobre l'ús i funcionament del motor gràfic i el renderitzat d'un joc per Android.
- Aprendre sobre l'ús i funcionament de llibreries de comunicació en xarxa i opcions multi-jugador com "Google Play Games Services".
- Aplicar conceptes adquirits durant la carrera en:
 - Fonaments de Programació
 - Programació Orientada a Objectes
 - Sistemes Operatius
 - Matemàtiques
 - Estadística
 - Interacció Humana amb els Ordinadors
 - Protocols i Aplicacions d'Internet
 - Administració de Xarxes i Sistemes Operatius
 - Gestió de Projectes

1.3 Enfocament i mètode seguit

Per elaborar un projecte d'aquestes característiques cal en primer lloc disposar d'un mínim de coneixements sobre el sistema operatiu Android i el funcionament a nivell de programació sobre aquest. Cal dir però, que a banda dels coneixements previs ha estat necessari assolir-ne molts altres de nous per poder dur a terme la implementació del joc. De totes maneres el primer pas abans de tractar a nivell de codi sempre és un disseny a nivell estructural del que es pretén realitzar.

Així, els apartats previs a aquesta documentació han estat fonamentals per poder desenvolupar completament la idea plantejada inicialment. Val a dir, que per un projecte d'aquestes característiques, és crucial plasmar la idea inicial sobre paper, elaborant esquemes i

descripcions de cada apartat, fet que es pot observar en aquesta documentació. D'aquesta manera la implementació resulta molt més ben estructurada des d'un primer moment.

Abans de començar però amb la implementació del codi, ha estat imprescindible una fase prèvia d'investigació. Així, els coneixements adquirits arrel del desenvolupament del projecte no han estat fruit de l'atzar ni l'espontaneïtat, ja que la planificació i el disseny previs del projecte han permès guiar les investigacions necessàries per obtenir el resultat final esperat, (tot i que, com es veurà més endavant, en aquesta fase d'investigació han sorgit alguns imprevistos que han fet rectificar el rumb previst inicialment).

D'aquesta manera, una vegada dissenyat i planificat el desenvolupament del projecte, cal començar amb la implementació seguint les pautes marcades. Pel que fa a la programació per Android, es poden seguir molts mètodes, però sempre es comença "dibuixant" quelcom en pantalla i posteriorment se'n programa el comportament. Si es tracta d'un joc, el fet de representar quelcom en pantalla requereix una atenció especial, ja que, sobretot en el transcurs d'una partida, no es tracta d'una representació estàndard i estàtica com es podria trobar en qualsevol aplicació senzilla, com serien camps per introduir text, botons, llistes, desplegable, etc. Així, en el cas d'un joc, la representació en pantalla de les imatges en moviment que corresponen a l'execució d'una partida, es duu a terme pel motor gràfic del joc i és el que s'anomena renderitzat.

Aleshores, es pot separar el desenvolupament del joc en dos apartats, el desenvolupament de l'entorn del joc, és a dir fora de la partida i el desenvolupament del joc en si, és a dir la pròpia partida. Pel que fa a l'entorn del joc, els menús inicials i les pantalles d'accés previs a una partida, es segueix el mètode normal de programació per Android, es dissenya els Layouts corresponents a cada pantalla ja establerta en la fase de disseny i posteriorment es programa el comportament de cada element de cada pantalla per tal que respongui correctament a les ordres de l'usuari.

Pel que fa al desenvolupament de la partida del joc es disposa de diferents alternatives a seguir, com ara utilitzar frameworks externs o seguir la documentació de Google per utilitzar les eines del propi SDK d'Android per representar les imatges mitjançant el que s'anomena Canvas. El Canvas és com la tela d'un pintor on es pot dibuixar directament, fet que té les seves avantatges i desavantatges, però principalment és el mètode que proposa Google per defecte.

En primer lloc val a dir que mitjançant el Canvas, es pot renderitzar qualsevol cosa però pot resultar feixuc d'utilitzar si el que es vol representar és un joc relativament complex, ja que cal redibuixar tota la pantalla a cada fotograma del joc, essent necessari controlar molts aspectes de forma manual a banda de la lògica de la partida en si. Per altra banda, mitjançant algun Framework extern es pot simplificar les tasques de desenvolupament, permetent una implementació en un nivell superior, programant les reaccions i interaccions de cada objecte que apareix en pantalla, abstraient-se de la lògica de renderitzat i permetent així al programador centrar-se en la lògica del joc i no tant en el motor gràfic.

En segon lloc, és important tenir clar que si bé un Framework gràfic pot aportar moltes avantatges pel que fa a la simplificació de la programació d'un joc, aquest requereix d'un aprenentatge previ, ja que, tot i que cada Framework pot utilitzar un llenguatge de programació estàndard (com per exemple Java, C#, etc.), els mètodes i la estructura d'un Framework concret sempre són únics, i per tant cal aprendre prèviament el funcionament a nivell de programació i estructura del Framework que es decideixi utilitzar.

Així, per realitzar aquest projecte inicialment s'ha escollit un Framework per la representació de gràfics i permetre centrar-se principalment en el desenvolupament del joc. De totes maneres, com es veurà més endavant, l'aprenentatge d'un nou Framework representa una corba d'aprenentatge i un cost temporal molt elevat, fet que ha suposat tenir que replantejar la estratègia inicial i reprendre el mètode clàssic de representació gràfica mitjançant el Canvas.

Per altra banda, cal plantejar com realitzar l'apartat multi-jugador, i en aquest cas es troba una situació similar a la del motor gràfic del joc, és a dir, es pot utilitzar les eines pròpies del SDK d'Android per establir comunicacions amb un servidor extern via Sockets i gestionar des d'aquest les diferents connexions i peticions per gestionar les partides multi-jugador, o per altra banda es pot fer ús de llibreries externes de serveis de tercers, que permeten una abstracció de la capa de comunicacions i centrar-se amb la lògica de negoci del joc per establir i gestionar una partida multi-jugador. Per la realització d'aquest projecte s'ha escollit fer ús de les llibreries de "Google Play Games Services" per gestionar l'apartat multi-jugador, aquestes s'exposen en apartats posteriors d'aquesta documentació.

Finalment, una vegada escollits els diferents mètodes a seguir per la implementació del joc, es pot plantejar la estratègia a seguir per poder implementar els diferents apartats d'aquest projecte. Així, el primer pas és analitzar cada apartat plantejat i investigar sobre el seu funcionament, sempre seguint la primícia de "no reinventar la roda" sinó que "fer-la girar de forma particular", ja que es considera més important aprendre i entendre el funcionament a nivell de programació de cada apartat, aprofitant les eines que ja es disposa, que crear-ne de noves des de zero desaprofitant els recursos i el temps disponible.

Així, durant la investigació prèvia no només s'ha estudiat el funcionament de cada apartat a realitzar, sinó que s'ha après d'exemples alternatius al que es vol realitzar, i en certa mesura s'ha aprofitat codi o lògica dels diferents exemples analitzats o inclús de tasques personals prèvies a la realització d'aquest projecte.

1.3.1 Recursos inicials

“El Gran libro de Android”[1]

Es tracta del llibre de text corresponent al curs de programació d'Android impartit per la Universitat Politècnica de València, redactat per el propi professor del curs. En aquest llibre es descriu àmpliament tot el sistema de programació per Android des de la base fins a conceptes força avançats. Es tracta cada apartat amb explicacions teòriques complementades d'exemples pràctics, fet que permet una bona comprensió de la base de la programació per Android. El temari del llibre comença per la configuració de l'entorn de programació (Eclipse), segueix per les funcionalitats més bàsiques de la programació per Android, com la implementació de la interfície gràfica, la interacció entre activitats o pantalles, la implementació d'escoltadors, llistes, etc. Avançant en aspectes més complexos com l'emmagatzematge de dades persistent, les llibreries gràfiques pròpies d'Android, les connexions a servidors externs a través de sockets, etc.

Aquest recurs pot ser molt important per al desenvolupament del projecte, ja que tracta a grans trets molts dels aspectes que seran necessaris per la implementació del projecte. Lluny de ser un recurs molt específic, permetrà consultar detalls en moments concrets i podrà aportar solucions a possibles problemes que puguin sorgir durant la programació o disseny de l'estructura del projecte. En ser un llibre ja estudiat anteriorment, no resultarà complicat trobar els punts necessaris a mode d'exemple de forma ràpida i funcional.

Documentació oficial d'Android [2]

Es tracta de la documentació oficial d'Android proporcionada directament de la ma de Google i disponible per a qualsevol programador d'Android. No està preparada per aprendre des de zero, però una vegada enteses les bases de la programació orientada a objectes amb Java, i les bases de la programació per Android, ens aporta informació sobre totes les llibreries pròpies del sistema i sobre el funcionament en sí de tot el que permet l'SDK d'Android. Gran part de la documentació es basa en explicacions sobre el funcionament a nivell de programació, i una altra part es basa en el detall de les llibreries i mètodes disponibles per la programació.

Aquest recurs pot resultar molt important en l'elaboració del projecte, com a font de consulta esporàdica, ja que resulta impossible conèixer absolutament tots els mètodes i funcions disponibles de memòria. Per altra banda, aquest recurs queda implementat amb l'entorn de programació (Eclipse) mitjançant el plugin d'eines de desenvolupament de Android instal·lable a l'entorn (ADT, Android Development Tools). D'aquesta manera permet una consulta instantània des del propi entorn (Eclipse), fet que facilita molt les tasques de programació i comprensió de nous mètodes o llibreries.

Edu4Java[3]

En aquesta web es tracta molts aspectes relacionats amb la programació amb llenguatge Java mitjançant fragments de codi a tall d'exemple, explicacions i vídeo-tutorials, per tant tracta també sobre programació per Android. Hi ha varis apartats, classificats per temàtiques com per exemple: "Java bàsic", "Programació de jocs senzills amb Java", "Programació per Android", "Programació de jocs en Android", "Java Web Servlet, JSP" entre altres. No és un recurs amb un nivell molt avançat, però sí molt pràctic pel que fa a l'abast dels conceptes tractats, de manera que no es profunditza molt però es manté un ventall força extens de conceptes interessants. La metodologia és mitjançant vídeos explicatius d'exemples concrets, amb fragments de codi i textos que serveixen de guia o d'aclariments. Pel que fa a la programació de jocs per Android, es centra amb les llibreries pròpies d'Android per al desenvolupament, evitant així l'ús de llibreries externes.

Respecte al projecte, aquest recurs pot aportar un complement a la documentació oficial d'Android que com s'ha vist, també es contempla com a recurs. D'aquesta manera, es pot tenir una visió més pràctica pel que fa a les llibreries gràfiques disponibles en el propi sistema Android (a banda de conceptes bàsics de Java o del propi desenvolupament per Android). La metodologia utilitzada (vídeos) permet un punt de vista diferent pel que fa a la comprensió del desenvolupament de jocs per Android, ja que sovint resulta més efectiva una explicació verbal acompanyada d'imatges que una descripció teòrica pura. Com que encara no se sap si es desenvoluparà l'apartat gràfic i motor del joc mitjançant les llibreries pròpies del sistema o mitjançant llibreries externes, pot resultar interessant per poder comparar i poder triar un sistema o altre amb arguments sòlids segons el que resulti més còmode, pràctic o eficient en el cas que ens contempla.

LibGDX – Steigert - Android Development [4]

Blog dedicat a la programació per Android, concretament a l'ús de les llibreries gràfiques LibGdx. Aquest blog està enllaçat directament des de la web oficial de la llibreria LibGdx recomanat com a documentació sobre la llibreria gràfica en qüestió. En aquest recurs es detalla en forma de guies el funcionament bàsic de la llibreria gràfica, començant pels conceptes més fonamentals, i ampliant amb funcions més avançades com el control de joc, les animacions en dues dimensions, les classes bàsiques, etc. Per altra banda, no es tracta d'una documentació íntegrament teòrica, sinó que procura explicar com utilitzar la llibreria més que descriure-la.

En aquest projecte pot resultar interessant per entendre i comparar la llibreria gràfica LibGdx amb les llibreries pròpies de l'entorn de programació d'Android. De la mateixa manera que s'ha comentat amb el recurs anterior, abans de triar una llibreria o altra serà interessant verificar què ens pot resultar més eficient en el nostre cas, i per aquesta raó precisament s'ha

triat aquest recurs, ja que no es tracta d'un recurs purament teòric, sinó que permet observar i entendre el funcionament de la llibreria des de la base, però amb un enfocament molt pràctic.

Documentació oficial de Google [5]

Es tracta de la documentació oficial de Google per als programadors. Aquest recurs és molt ampli, i nosaltres ens centrarem amb els apartats de desenvolupament de jocs, on es pot trobar per una banda la secció de serveis per jocs (Game services), que es basa en oferir serveis al núvol per al desenvolupament de jocs, de manera que es permet l'ús de la infraestructura de Google per l'emmagatzematge d'informació de cada usuari respecte al progrés en el joc, les puntuacions, etc. Aquest servei també permet la implementació de capacitats multi-jugador i l'assignació de missions en un mode d'èxits del jugador compartibles a través de la xarxa social de Google (Google +). Per altra banda, aquest recurs també disposa d'una secció referent a "Google App Engine", que permet executar aplicacions web amb llenguatge Java sobre la infraestructura de Google sense necessitat d'utilitzar servidors externs.

Aquest recurs, tot i ser molt extens pot ser molt interessant per la implementació del mode multi-jugador al joc. Com s'ha comentat, es pretén utilitzar sockets a partir d'un servidor extern, i una opció podria ser utilitzar App Engine per poder utilitzar els servidors de Google i evitar així costos en lloguer de servidor. Per altra banda, com que el llenguatge que suporta és Java, ens permetria crear el programa servidor mitjançant sockets amb llenguatge Java, així, pot ser un recurs força interessant en aquest àmbit. En segon lloc, es pot comparar la implementació mitjançant sockets per les tasques de multi-jugador, amb l'ús de les API de "Google Play Game Services", que poden aportar unes capacitats i fiabilitat superiors a les dels sockets, tot incloent capacitats socials al projecte. D'aquesta manera pot resultar interessant ja sigui per comparar les dues alternatives, com per agafar un punt de vista de les metodologies utilitzades actualment pel que fa al desenvolupament de jocs per Android amb capacitats socials i multi-jugador.

1.4 Planificació del Treball

Per a tot projecte és primordial una planificació temporal prèvia de les tasques a realitzar per tal d'assolir els objectius establerts inicialment. Així, en aquest apartat es mostra la planificació proposada, val a dir però, que com ja s'ha comentat en apartats anteriors, el projecte ha sofert alguna desviació pel que fa al plantejament inicial degut a determinats contratemps que han implicat alguna modificació sobre la primera proposta, per això en aquest apartat es mostra a banda de la planificació inicial, una descripció sobre la adaptació a la situació real.

Detall de la planificació temporal













































		Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin
1			PAC 1: Proposta de Projecte	11 días	mié 18/09/13	mar 01/10/13
2			PAC 2: Estat de l'Art	15 días	jue 03/10/13	lun 21/10/13
3			⊖ PAC 3: Disseny i implementació	38,25 días	jue 24/10/13	dom 15/12/13
4			Planificació	2 días	jue 24/10/13	vie 25/10/13
5			⊖ Disseny	13,63 días	vie 25/10/13	mar 12/11/13
6			Definició Estructura Aplicació	13 días	vie 25/10/13	dom 10/11/13
7			Definició Funcionalitats del Joc	2 días	sáb 26/10/13	dom 27/10/13
8			⊖ Definició de Partida	3,38 días	dom 27/10/13	vie 01/11/13
9			Partida: Definició funcionament	2 días	dom 27/10/13	mar 29/10/13
10			Partida: Definició Puntuacions	2 días	lun 28/10/13	mar 29/10/13
11			Definició Funcionalitat Multi-Jugador	3 días	mié 30/10/13	sáb 02/11/13
12			⊖ Investigació Motor Gràfic	3,13 días	sáb 02/11/13	mar 05/11/13
13			Motor Gràfic: Investigar possibilitats	2 días	sáb 02/11/13	dom 03/11/13
14			Motor Gràfic: Escollir Framework	1,75 días	lun 04/11/13	mar 05/11/13
15			⊖ Investigació Llibreries Externes i possibilitats	2,5 días	mié 06/11/13	sáb 09/11/13
16			Llibreries Externes: Investigar possibilitats	3 días	mié 06/11/13	sáb 09/11/13
17			Llibreries Externes: Definir Funcionalitats extra a incorporar	2 días	vie 08/11/13	sáb 09/11/13
18			Investigació Multi-Jugador	3 días	sáb 09/11/13	lun 11/11/13
19			⊖ Implementació	24,88 días	mié 13/11/13	dom 15/12/13
20			⊖ Aplicació/Entorn del Joc	3,63 días	mié 13/11/13	sáb 16/11/13
21			Layouts	2,25 días	mié 13/11/13	vie 15/11/13
22			Desenvolupament Activitats	3,13 días	jue 14/11/13	sáb 16/11/13
23			⊖ Investigació/Formació Framework gràfic triat	8,13 días	sáb 16/11/13	mar 26/11/13
24			Framework: Documentació	2,63 días	sáb 16/11/13	lun 18/11/13
25			Framework: Exemples	3,25 días	lun 18/11/13	vie 22/11/13
26			Framework: Desenvolupament Partida del Joc	5,75 días	mié 20/11/13	mar 26/11/13
27			Framework: UATs funcionament Partida	1,88 días	dom 24/11/13	mar 26/11/13
28			⊖ Investigació/Formació Llibreries externes	3,63 días	mié 27/11/13	sáb 30/11/13
29			Llibreries Externes: Documentació	1,75 días	mié 27/11/13	jue 28/11/13
30			Llibreries Externes: Exemples	3,13 días	jue 28/11/13	sáb 30/11/13
31			Llibreries Externes: Desenvolupament Funcionalitats Socials Extra	2 días	vie 29/11/13	sáb 30/11/13
32			Llibreries Externes: UATs funcionament Partida	1 día	sáb 30/11/13	sáb 30/11/13
33			⊖ Investigació/Formació Multi-Jugador	8,13 días	sáb 30/11/13	mar 10/12/13
34			Multi-Jugador: Documentació	2,25 días	sáb 30/11/13	dom 01/12/13
35			Multi-Jugador: Exemples	3,38 días	dom 01/12/13	vie 06/12/13
36			Multi-Jugador: Desenvolupament Partida Multi-Jugador	6,25 días	mar 03/12/13	mar 10/12/13
37			Multi-Jugador: UATs funcionament Partida Multi-Jugador	1,88 días	dom 08/12/13	mar 10/12/13
38			⊖ Documentació	38,25 días	jue 24/10/13	dom 15/12/13
39			Esborranys i esquemes sobre la feina realitzada	35,25 días	jue 24/10/13	mar 10/12/13
40			Agrupació de documentació generada i "Pas a Net" per Entrega	5,38 días	mar 10/12/13	dom 15/12/13
41			PAC 4: Memòria	18 días	lun 16/12/13	jue 09/01/14
42			PAC 5A: Presentació	7 días	jue 09/01/14	vie 17/01/14
43			PAC 5B: Defensa	7 días	sáb 18/01/14	dom 26/01/14

Figura 1. Detall de la planificació temporal.

1.4.1 Diagrama de Gantt

PACs 1 i 2

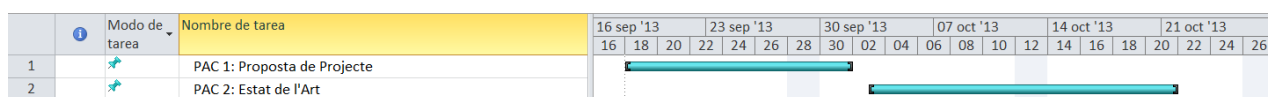


Figura 2. Diagrama de Gantt – PACs 1 i 2.

PAC 3

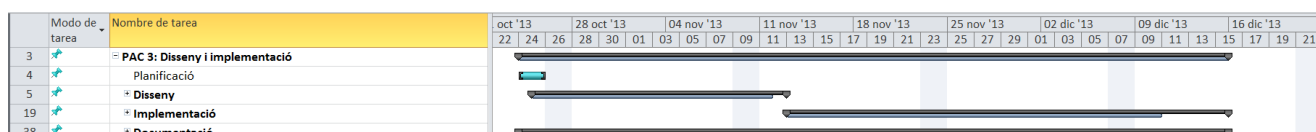


Figura 3. Diagrama de Gantt – PAC 3.

La tercera PAC correspon al disseny i implementació del projecte, el qual es divideix en dues fases principals i una tercera corresponent a la documentació de tot el procés. La primera fase correspon al disseny del joc, on es tracta de marcar les bases del que posteriorment es desenvolupa, estructurant i planificant el funcionament tant de l'entorn del joc com les partides i les funcionalitats d'aquest. La segona fase correspon a la implementació, on es programa el joc seguint les pautes que s'han marcat en la fase de disseny anterior.

PAC 3 – Detall Fase Disseny

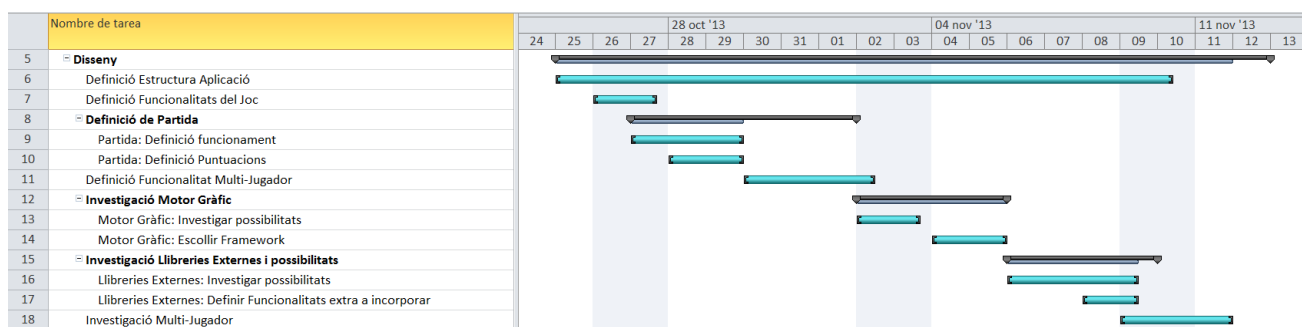


Figura 4. Diagrama de Gantt – Detall fase de disseny.

Durant el disseny, cal definir tant la estructura de l'aplicació que conté el joc com el funcionament de les partides d'aquest. Així, per definir la estructura general de l'aplicació, cal establir les diferents funcionalitats del joc i el seu entorn i per tant el disseny de la estructura de l'aplicació s'allarga durant pràcticament tota la fase de disseny.

Així, en aquesta fase cal definir a més, les diferents funcionalitats que ha de permetre el joc i establir les normes pel que fa al transcurs de cada partida, definint alhora el mètode de puntuacions a establir. Per altra banda, tenint en compte que es tracta d'un joc multi-jugador, en aquesta fase cal establir les bases del comportament durant les partides multi-jugador.

En aquesta fase és necessari decidir com programar posteriorment els diferents apartats del joc, com el motor gràfic d'aquest, les possibles llibreries externes a utilitzar per implementar funcionalitats extra o les alternatives per desenvolupar l'apartat multi-jugador.

PAC 3 – Detall Fase Implementació

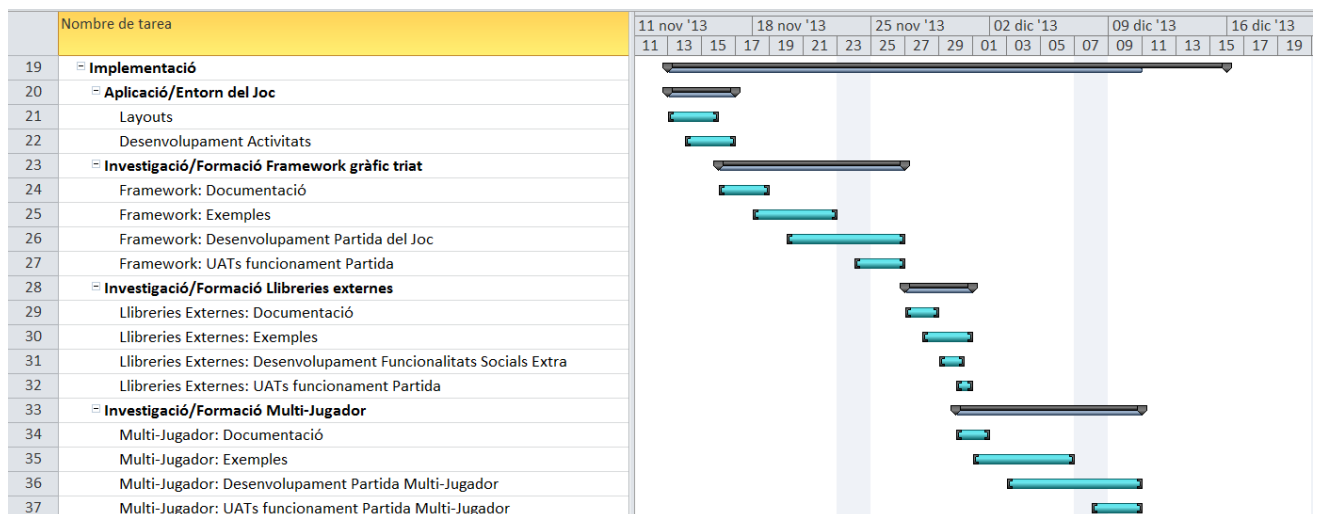


Figura 5. Diagrama de Gantt – Detall fase d'implementació.

En la fase d'implementació cal plasmar sobre codi tot el que s'ha establert en l'apartat anterior, per tant es comença implementant la estructura de la aplicació que conté el joc seguint les bases del desenvolupament per Android, implementant en primer lloc la interfície gràfica de cada activitat (Layouts) i en segon lloc enllaçant aquestes al codi per poder programar-ne el funcionament i les interaccions dissenyades.

Una vegada es disposa de la estructura, es comença amb la implementació del joc en si, és a dir, la activitat que correspon a la inicialització d'una partida. Però aquesta no es desenvolupa de la mateixa manera que una activitat normal de qualsevol aplicació Android, sinó que cal implementar-la amb un motor gràfic, que és l'encarregat de renderitzar i gestionar el funcionament de la partida. En la fase de disseny s'ha escollit un Framework a utilitzar per desenvolupar aquest apartat, i per això cal primer aprendre a utilitzar-lo. Per tant, els primers dies cal estudiar la documentació disponible sobre aquest, fet que deriva en l'anàlisi dels exemples disponibles per la millor comprensió del Framework a nivell de programació.

Així, finalment gràcies a la documentació i els exemples es pot començar amb el desenvolupament del joc pròpiament dit, fet que implica les corresponents UATs (User

Acceptance Tests) per validar que el funcionament i comportament del joc és l'esperat. Val a dir que aquestes proves finals impliquen accions i correccions a nivell de codi per tal d'aconseguir el resultat esperat.

Posteriorment, s'aplica el mateix procediment per implementar les funcionalitats extra mitjançant les llibreries externes escollides en la fase de disseny i per l'apartat multi-jugador. És a dir, en primer lloc s'estudia la documentació i exemples disponibles, i en segon lloc es programa seguint el que s'ha dissenyat a partir dels coneixements adquirits durant l'estudi, realitzant les proves i validacions corresponents en cada apartat.

Val a dir, que durant la fase d'implementació s'ha sofert determinats contratemps que no estaven planificats, que han implicat un replantejament a nivell de disseny i han afectat a la temporització del projecte, aquests es detallen posteriorment en aquest mateix apartat.

PAC 3 – Detall Documentació

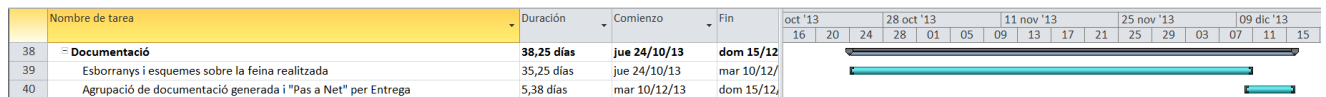


Figura 6. Diagrama de Gantt – Documentació

Paral·lelament al disseny i la implementació del joc, cal documentar els procediments i les decisions preses en cada fase. Per aquesta raó, durant tot l'apartat cal elaborar esborranys i esquemes, per finalment poder crear la documentació corresponent.

PACs 4 i 5

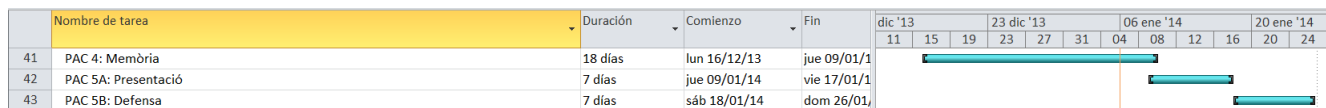


Figura 7. Diagrama de Gantt – PACs 4 i 5.

Finalment, cal elaborar la memòria del projecte, on s'agrupa tot el que s'ha realitzat durant el projecte en una única documentació final. Així, posteriorment cal elaborar una presentació del projecte final i defensar-lo virtualment davant el tribunal corresponent.

1.4.2 Desviació

Tal com s'ha comentat anteriorment, durant la fase de implementació s'ha decidit modificar la planificació i el disseny previst inicialment, ja que es considera que es va cometre un error al escollir el Framework per implementar el motor gràfic del joc.

Així, durant el disseny d'aquest punt, després d'analitzar les diferents alternatives entre les quals hi havia LibGDX, AndEngine o les eines del propi SDK d'Android a través de la classe Canvas, es va decidir utilitzar el Framework LibGDX. Aquesta decisió es va prendre degut al alt rendiment i les bones crítiques de la comunitat de programadors respecte aquest en comparació a les alternatives plantejades. Tot i estar informat de la complexitat d'aquest i la dificultat implícita a la gran corba d'aprenentatge que suposa utilitzar LibGDX per primera vegada, es va cometre l'error de pensar que amb la temporització planificada seria suficient per obtenir els resultats esperats.

Aquest fet va suposar un greu error en el plantejament inicial, sobretot pel que fa a la temporització del desenvolupament, ja que es va perdre un temps molt valuós investigant i aprenent sobre el funcionament i la implementació mitjançant aquest Framework que finalment es va descartar.

Davant la situació en que el temps estipulat per la implementació del motor gràfic s'exhauria i encara no es dominava ni entenia el funcionament de LibGDX es va decidir fer un pas endarrere i agafar un nou camí més realista tenint en compte el temps restant per poder assolir els objectius establerts inicialment. Així, es va tenir que replantejar el disseny del motor gràfic del joc, i gràcies a les investigacions dutes a terme en la fase de disseny, la elecció va ser ràpida, escollint utilitzar les eines del propi SDK d'Android i descartant definitivament LibGDX.

De totes maneres ha estat necessari aprendre les bases del funcionament del motor gràfic d'un joc per tal d'utilitzar la classe Canvas d'Android. Tot i així, la corba d'aprenentatge no ha estat tant elevada com en el cas de LibGDX, però el grau de dificultat no ha disminuït excessivament, de totes maneres, els coneixements previs sobre la programació per Android han permès un aprenentatge suficientment àgil per arribar a implementar el motor gràfic del joc mitjançant el Canvas d'Android respectant de forma relativament acurada la temporització inicial (escurçant parcialment les temporitzacions de les fases següents).

La part positiva d'aquesta errada és que a banda d'aprendre un mínim sobre LibGDX, (fet que no es descarta poder aplicar en un futur de cara a altres projectes personals), ha estat la capacitat de reacció adquirida en el procés, ja que en observar un possible des-ajustament sobre el pla previst, amb la possibilitat de no respectar els objectius fixats, es va saber reaccionar i adoptar una alternativa més viable mantenint els criteris inicials.

1.5 Breu sumari de productes obtinguts

En acabar el projecte s'ha aconseguit crear un joc del tipus "Shoot'em Up" per Android amb capacitat per realitzar partides multi-jugador, i funcionalitats socials com poder comparar les puntuacions obtingudes per cada usuari amb la xarxa social de Google.

Tot seguit es mostra el joc realitzat mitjançant captures de pantalla.

Pantalla inicial, sense iniciar sessió a "Google +" i una vegada iniciada la sessió:

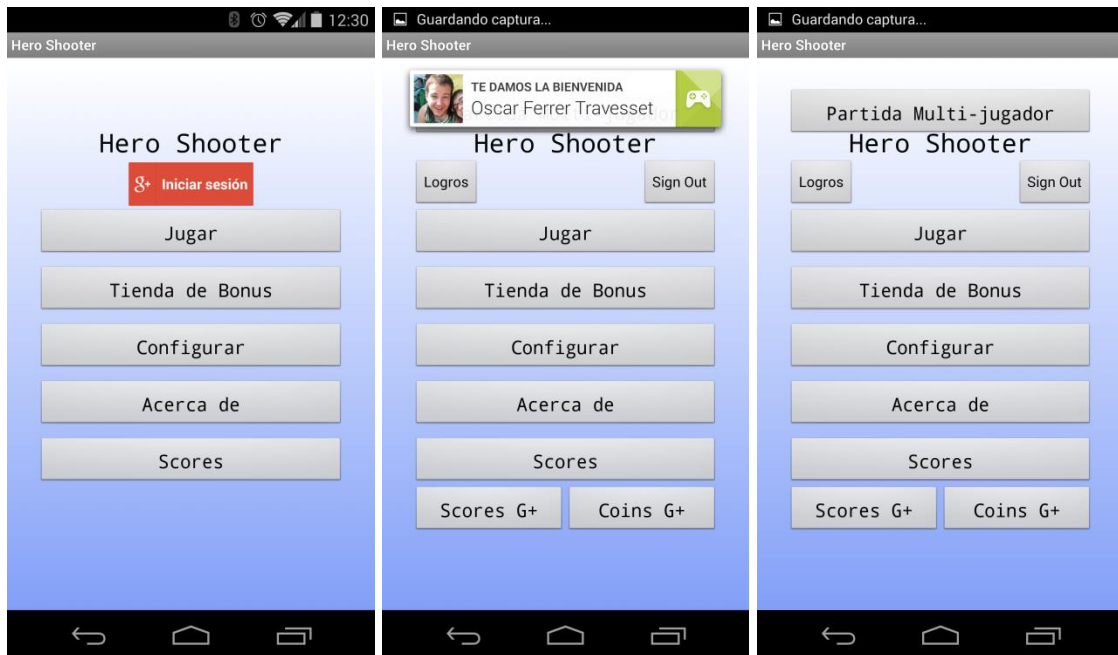


Figura 8. Captures de pantalla – Pantalla inicial.

Fites i puntuacions, en mode local i compartides mitjançant el compte de "Google +":

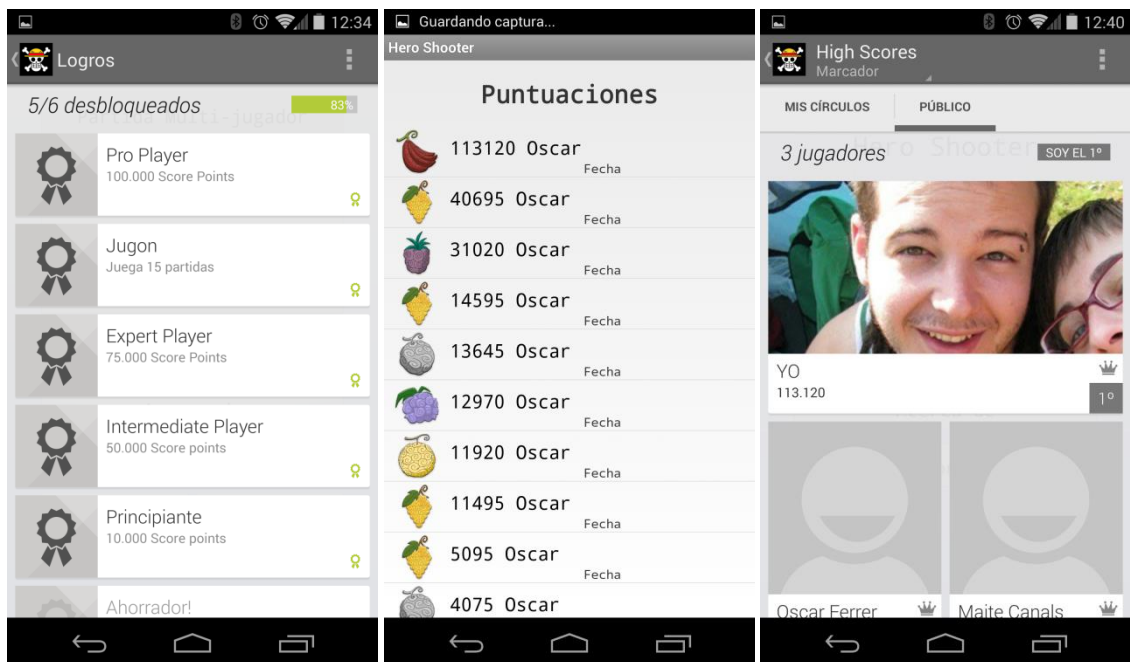


Figura 9. Captures de pantalla – Fites i puntuacions.

Botiga de bons i mode multi-jugador, (pantalla inicial i cerca automàtica d'oponents - Quick Game):

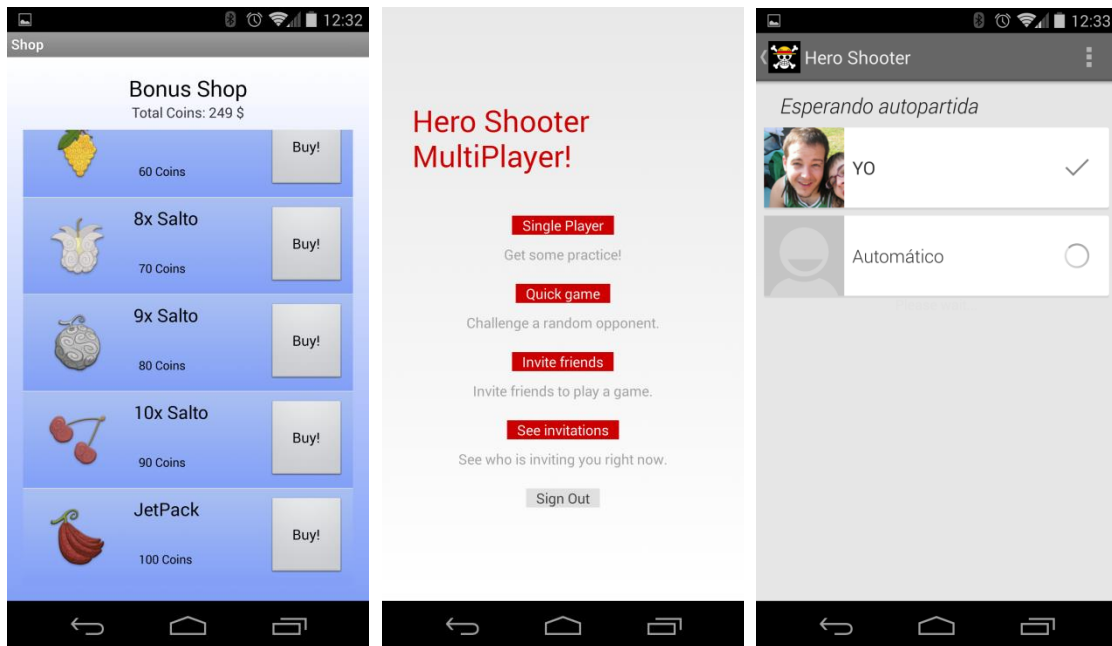


Figura 10. Captures de pantalla – Botiga de bons i multi-jugador.

Transcurs d'una partida:

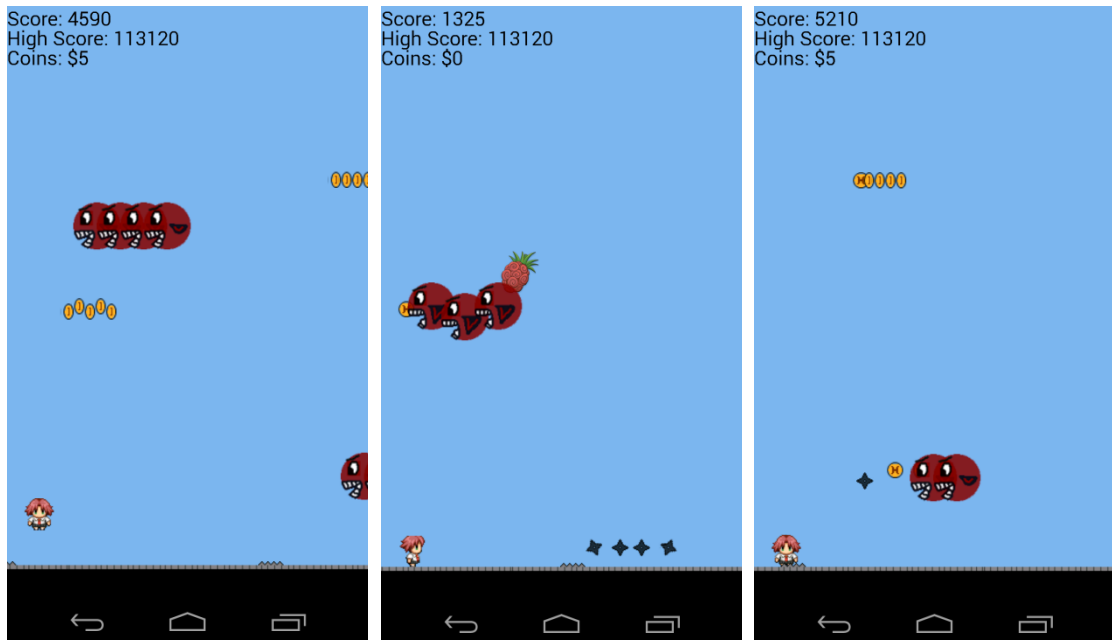


Figura 11. Captures de pantalla - Partida.

2. Estat de l'art

2.1 Telèfons mòbils intel·ligents: l'aparició dels "Smartphones"

Els primers telèfons mòbils intel·ligents van aparèixer als voltants de l'any 2003 amb l'aparició de les primeres "agendes electròniques" (PDAs) amb capacitat per efectuar trucades, enviar missatges SMS i consultar el correu electrònic, tot a través d'una pantalla tàctil. La primera empresa en desenvolupar aquests dispositius va ser la coneguda PALM, que ja des dels voltants dels anys 90 fabricava PDAs (sigles del anglès "Personal Digital Assistant"), i que a partir de 2003 va dotar-los amb les capacitats de un telèfon mòbil. En català, inicialment es coneixien aquests dispositius com agendas electròniques i més endavant, com a ordenadors de butxaca (traducció directa de l'anglès "Pocket PC").

A mesura que va avançar el temps, la tecnologia va anar evolucionant, millorant les característiques de les PDAs tradicionals, i així també els sistemes operatius que utilitzaven. Així, al 1997 va aparèixer el sistema operatiu de Microsoft per a "Pocket PCs", anomenat "MS Windows CE" i al 2003 va ser millorat i re-nomenat com a "Microsoft Windows Mobile 2003", versió que seria actualitzada al 2005 amb el nom de "MS Windows Mobile 5.0" i al 2007 amb la versió "MS Windows Mobile 6", arribant fins a la versió 6.5 a mitjans del 2009. A partir del 2010, el sistema va realitzar un gir per adaptar-se al mercat general i no tant al mercat empresarial, re-nomenant-se així a "MS Windows Phone 7", evolucionant posteriorment amb les versions "7.5" i "7.8". Finalment, a finals del 2012 es va modificar el nucli del sistema per adaptar-lo al mercat actual, creant així la versió que es disposa en el present, "MS Windows Phone 8".

Aquests dispositius, des d'un inici rivalitzaven amb els dispositius "BlackBerry", que des del 2003 comercialitzava el seu propi telèfon intel·ligent orientat a executius i empresaris, on es diferenciaven dels "Pocket PC" (a part del sistema operatiu), en que aquests últims no disposaven de pantalles tàctils, al seu defecte oferien un teclat qwerty complet, miniaturitzat sobre el terminal. "BlackBerry" va ser creada per l'empresa RIM, (Research In Motion), i van gaudir d'un gran èxit sobretot en el món empresarial, principalment gràcies a la eficient gestió de correus electrònics des del telèfon i els anomenats "BlackBerry Services", que proporcionen una seguretat extra pel que fa a la connectivitat a internet, aportant serveis de valor afegit com el popularment conegut com "BlackBerry Messenger". Actualment, tot i sofrir dificultats per mantenir l'empresa a un nivell competitiu, segueix fabricant telèfons intel·ligents.

Paral·lelament a l'any 2007, "Apple inc." va comercialitzar el seu primer telèfon intel·ligent, que va anomenar "iPhone", i va suposar una revolució en el mercat, ja que els telèfons amb "Windows Mobile" no gaudien de gaire èxit i eren poc intuïtius de cara al públic general, per altra banda les "BlackBerry" estaven orientades a executius i no gaudien del beneplàcit del mercat domèstic. Així, va aparèixer el primer Smartphone per a tots els públics, que va ser un

gran èxit de vendes i va popularitzar els telèfons intel·ligents. Posteriorment, any rere any i fins al present, han millorat el dispositiu creant-ne un nou model de forma anual, passant per “iPhone 3G” al 2008, “3GS” al 2009, “4” al 2010, “4S” al 2011, “5” al 2012, “5C” i “5S” al 2013. Val a dir, que des del primer fins l’últim aquests dispositius són un dels grans referents en el món de la telefonia mòbil.

Arrel de l’èxit aconseguit per Apple amb el seu primer Smartphone, van ser molts els fabricants en intentar posar a la venda els seus propis telèfons mòbils intel·ligents, alguns amb sistemes operatius propis, com el “Bada OS” de Samsung, altres com Nokia i Sony Ericsson van optar per fer renéixer el sistema “Symbian OS”, que tradicionalment utilitzava Nokia per als seus mòbils convencionals, adaptant-lo a les capacitats d’un mòbil intel·ligent. De totes maneres, l’únic sistema que ha sigut capaç de plantar cara a les exigències del mercat des de l’aparició del primer iPhone va ser el sistema operatiu “Android”.

Android és un sistema operatiu per a mòbils basat en Linux que va ser desenvolupat per “Android Inc.” a l’any 2003. Però no va ser fins al 2005 que l’empresa va ser adquirida per Google per, al 2007 presentar-lo al públic general a través d’un consorci d’empreses anomenat “Open Handset Alliance” (OHA). De totes maneres, el primer mòbil comercialitzat amb el sistema operatiu Android no va sortir a la venda fins al 2008, un any més tard que el primer iPhone de Apple. Tot i arribar al mercat un any més tard, la OHA va saber recuperar l’avantatge presa per Apple utilitzant una estratègia diferent. Com hem vist, Apple es dedica tant a la confecció del hardware dels seus dispositius com el software, de manera que només disposa d’un sol dispositiu al qual adaptar el sistema operatiu. Amb aquesta estratègia, Apple va aconseguir un sistema molt polit, estable i fluid. Per contra la OHA, al tractar-se d’una aliança entre empreses va procurar obrir l’abast del producte de Google a qualsevol fabricant de mòbils intel·ligents. D’aquesta manera, tot i arribar un any més tard que el primer iPhone, el sistema operatiu Android va poder obrir-se camí en un ventall molt ampli de dispositius, i per tant arribar a una diversitat de públic major a la de Apple.

No obstant, la guerra entre Android i Apple és dura, de fet en el present encara segueix latent, i no només entre ambdós. Com s’ha comentat anteriorment, hi ha altres sistemes operatius per a telèfons intel·ligents, però durant la gran guerra de mercat per aconseguir estar a primera línia i convertir-se en el sistema més utilitzat per al públic general, molts dels sistemes operatius han caigut en desús. Els únics que en l’actualitat segueixen resistint són Microsoft amb el “Windows Phone 8”, RIM amb les “BlackBerry”, i els dos grans pilars actuals, Apple amb els “iPhone” i Google amb “Android”.

De totes maneres, RIM comença a donar signes de debilitat i sembla que poc a poc va perdent mercat, i possiblement acabi cedint el pas, mentre que Microsoft, va formar una aliança amb el fabricant Nokia per assegurar-se la presència del seu sistema operatiu en un dels fabricants de més renom abans de l’aparició dels Smartphones, i sembla que gràcies a aquest fet, poc a poc va agafant presència en aquest món.

Tot i així, podem concloure que en l'actualitat els dos sistemes més usats per al gran públic en general són el d'Apple i el de Google, amb la diferència que Android engloba un ventall molt ampli de dispositius, des de la gamma més bàsica o senzilla fins als dispositius de més alta gamma, mentre que Apple només disposa d'un dispositiu d'alta gamma. Per tant, en el sistema Android podem trobar un públic objectiu molt més ampli i variat que en el cas del sistema de Apple.

2.1.1 Usos freqüents

Tal com hem vist, l'arribada dels Smartphones ha suposat una revolució en el món de la telefonia mòbil. Potser en l'aparició dels primers terminals intel·ligents el context tecnològic i social encara no estava preparat per l'auge que va venir més tard. Així, la gran explosió d'aquesta revolució va sorgir juntament amb l'arribada massiva d'internet als dispositius mòbils a través de la xarxa 2G i posteriorment 3G. En aquell moment, els usuaris podien consultar el correu electrònic des de qualsevol indret amb el seu Smartphone, podia consultar internet per fer cerques o navegar per les xarxes socials. En aquest últim punt es pot constatar un auge de les xarxes socials a través dels dispositius mòbils gràcies a les aplicacions que ràpidament van aparèixer per cada plataforma mòbil.

Així, de forma subliminal acabem de veure quin va ser un dels punts d'inflexió que va permetre o millor dit, va significar la supervivència d'uns sistemes operatius i la mort dels altres. Com hem vist fins ara, els sistemes que han persistit en l'actualitat han estat 4, BlackBerry, Windows Phone, iPhone i Android, i el punt que tenen en comú és que tots permeten instal·lar aplicacions de tercers després d'adquirir el terminal per part de l'usuari. Així, un dels grans pilars que sustenta el mercat dels Smartphones són les aplicacions, popularment conegudes com a "apps" (de l'anglès application).

D'aquesta manera, cada plataforma o sistema operatiu disposa de la seva pròpia "botiga d'aplicacions" on els usuaris poden trobar i descarregar "apps" per afegir funcionalitats al dispositiu. Val a dir doncs, que la tecnologia no ho és tot, ja que els terminals disposen de prestacions suficients per realitzar pràcticament qualsevol cosa que es pugui dur a terme amb un ordinador (dins uns determinats límits), però degut a la gran varietat de perfils de consumidors del mercat, es permet la personalització de les funcions a realitzar amb el terminal a base d'aplicacions. Per tant, els sistemes operatius mòbils disposen de fabrica les funcionalitats bàsiques (correu electrònic, calendari, trucades, SMS, etc.) i cada usuari personalitza el seu terminal a base d'aplicacions.

Així, poc a poc amb el pas del temps, s'ha anat modificant la forma en que els usuaris fan servir el telèfon mòbil, adoptant un caire més lúdic, fet que no es donava abans de l'aparició dels Smartphones i les aplicacions. D'aquesta manera, a banda dels telèfons mòbils intel·ligents van aparèixer nous dispositius per absorbir o per diversificar part de la vessant més lúdica dels

usos dels Smartphones, aquests són les tauletes digitals, popularment conegudes pel seu nom en anglès, "Tablets". Aquests dispositius són com un Smartphone, en un format més gran (entre 7 i 10 polzades de pantalla tàctil), però sense la funcionalitat de telefonia. De totes maneres, les tauletes es troben amb els mateixos sistemes operatius que els Smartphones, permetent la sincronització de dades entre els dispositius, ampliant així la compatibilitat i el grau de retroacció amb el terminal mòbil.

Per tant, ja sigui des d'un Smartphone o una tauleta, els usuaris utilitzen els seus dispositius majoritàriament per enviar o rebre correus electrònics, gestionar l'agenda personal, escoltar música, navegar per internet (ja sigui per realitzar cerques esporàdiques com per consultar les webs habituals de cada usuari), navegar per les xarxes socials, fer fotografies i bàsicament per activitats d'oci. Aquestes activitats relacionades amb l'oci poden englobar molts àmbits, ja sigui per les diverses aplicacions que es poden trobar a les diferents botigues d'aplicacions de cada plataforma, com per exemple, realitzar recorreguts a peu i mostrar-los sobre un mapa, controlar els hàbits alimentaris, gestionar les despeses personals, i una infinitat d'utilitats en tots els àmbits possibles, o per altra banda els usuaris es poden decantar directament per jugar a jocs des dels propis dispositius, aquests al igual que les aplicacions, es poden descarregar des de les respectives botigues d'aplicacions.

2.1.2 Capacitats per jugar

Com ja hem comentat, cada vegada més s'utilitza l'Smartphone per raons d'oci, i en aquest sentit cada vegada són més les persones que utilitzen els dispositius mòbils per jugar. Així, les característiques de hardware d'aquests dispositius han evolucionat molt al llarg del temps tenint en compte aquesta premissa, desenvolupant processadors gràfics molt potents. De fet, han aparegut al mercat inclús vídeo consoles amb un sistema operatiu mòbil com per exemple la "Ouya" amb el sistema Android.

Paral·lelament, les botigues d'aplicacions de cada un dels sistemes operatius, s'han vist inundades de jocs per descarregar als dispositius corresponents, obrint pas a un gran mercat de jocs combinat amb el de les aplicacions. Es poden trobar doncs, jocs molt senzills, com petits puzles, jocs de preguntes, etc. Pensats per emplenar els buits de temps que pot trobar l'usuari en el seu dia a dia, com esperant l'autobús, fent cua al metge, etc. Per altra banda, es poden trobar jocs molt elaborats, amb càrregues gràfiques molt importants, en tres dimensions, o inclús jocs multi-jugador. L'avantatge és que cada usuari pot escollir els que vulgui i trobar el moment adequat per cada un, però no s'haurà de preocupar per qüestions tecnològiques, ja que les botigues d'aplicacions només mostraran el que sigui compatible amb el dispositiu en concret.

2.2 Sistemes Operatius Mòbils

En aquest apartat s'analitzarà per sobre les característiques de cada sistema operatiu mòbil, revisant les versions anteriors i centrant-nos amb la versió actual. Per altra banda es comentarà la situació actual de cada botiga d'aplicacions i el seu funcionament.

2.2.1 iOS

Els Smartphones i tauletes de Apple (iPhone i iPad) utilitzen el sistema operatiu del propi fabricant, aquest s'anomena iOS, que anteriorment s'anomenava "iPhone Operating System" (nom que va mantenir fins al llançament de l'iPhone 4 al 2010). La principal característica d'aquest sistema operatiu és que només es troba disponible per dispositius fabricats per Apple.

Inicialment, el primer iPhone portava instal·lada la versió 1.0, amb el segon (iPhone 3G) la v2.0, on la principal novetat va ser l'aparició de l'AppStore, la botiga d'aplicacions d'Apple, de la qual en parlarem més endavant. Així en sortir al mercat el següent model d'iPhone (3GS), es va avançar a la v3.0, que va introduir poques millores, entre altres per exemple la brúixola. En arribar al iPhone 4, va aparèixer la versió de iOS 4, que va aportar grans millores a l'ecosistema de Apple, les més importants es podrien resumir amb la capacitat per multi-tasques, el "Game Center" (plataforma per a jocs) i l'aparició de "Face Time", la plataforma del fabricant per realitzar vídeo-trucades. Amb el iPhone 4S va arribar el iOS 5, que permetia ser actualitzat per els models anteriors a partir del 3GS. Amb aquesta versió va sorgir el famós assistent de veu que es va anomenar "Siri", que permet realitzar consultes a través de la veu directament al mòbil. Per tant, amb el iPhone 5 va arribar el iOS 6, que entre altres novetats aportava uns nous mapes creats per Apple, per a la funcionalitat de GPS, on fins aleshores s'emprava Google Maps. Finalment, amb el iPhone 5C i 5S ha arribat iOS 7, que principalment aporta millores en el rendiment i la estabilitat del sistema, millorant la multi-tasca, el servei de notifikacions entre altres moltes. Val a dir però, que cada versió, a part de les característiques pròpies, també ha millorat l'apartat visual del sistema, creant en un conjunt un dels sistemes operatius mòbils més fluids, intuïtiu i fàcil d'utilitzar sense cap coneixement previ.

La botiga d'aplicacions, anomenada AppStore mereix una menció especial, ja que sempre ha gaudit del millor èxit possible, ja sigui per nombre d'aplicacions disponibles, com per la seguretat d'aquestes. Actualment però, ja no és la botiga amb més aplicacions, però segueix essent la més segura, tot seguit s'analitzarà el perquè.

Primerament, cal constatar l'immens nombre d'aplicacions disponibles, que a finals de setembre de 2013 es situa al voltant de 950.000 aplicacions disponibles.

En segon lloc, cal remarcar el nivell de seguretat de totes aquestes aplicacions, i aquest s'aconsegueix controlant cada aplicació que es publica a la botiga. Per poder ser programador

per iOS, en primer lloc és imprescindible disposar d'un ordinador Apple (amb el sistema operatiu "Mac OSX") on s'utilitza l'entorn de programació "X-Code", per programar amb llenguatge "Objective-C". Finalment, per poder publicar aplicacions, cal tenir un compte de programador de Apple, el qual té un cost anual de 100\$ (uns 70 euros al canvi). Per altra banda, una vegada es vol publicar una aplicació a la botiga, aquesta ha d'estar validada prèviament per la pròpia Apple, i una cop rebut el vist i plau (poden passar varis dies), la aplicació ja queda publicada a la botiga. D'aquesta manera s'evita la publicació d'aplicacions fraudulentas o possibles aplicacions malicioses, augmentant el nivell de seguretat de la botiga de cara als usuaris finals.

2.2.2 Windows Phone

Com ja s'ha vist anteriorment, el sistema operatiu per mòbils de Microsoft ha evolucionat molt al llarg del temps, amb les primeres versions dissenyades per a PDAs, amb varis canvis de noms, però fins al 2010 no va arribar el que es coneix com a Windows Phone, un sistema operatiu per Smartphones. Inicialment va ser la versió 7 i posteriorment la versió 7.5. En anunciar que la futura versió Windows Phone 8 no seria compatible amb els terminals anteriors que utilitzessin la versió 7, Microsoft va actualitzar la versió anterior a la 7.8 per dotar-la de alguna de les millores introduïdes en la futura versió (la actual versió 8). Així, amb Windows Phone 8 es va crear de nou el nucli del sistema, que entre altres millores en l'estabilitat del sistema, permet la multi-tasca per aplicacions de tercers, fet que en les versions anteriors era impossible.

La botiga d'aplicacions d'aquest sistema operatiu s'anomena Marketplace, que està disponible des del canvi de nom del sistema operatiu a Windows Phone. Aquesta botiga, en arribar més tard que els seus competidors i pel fet que és un sistema que li ha costat molt d'esforç arribar al gran públic, tot i ser un sistema amb relativament poca quota de mercat, al febrer de 2013 ja superava les 130.000 aplicacions.

L'entorn de programació que s'utilitza en aquest sistema és el VisualStudio, d'on s'utilitza l'SDK Silverlight i l'SDK Windows Phone. A partir d'aquí, el llenguatge de programació a utilitzar és el C#. Per altra banda, per poder publicar aplicacions cal registrar-se com a programador amb un cost anual de 75 euros. Per altra banda, i al igual que en el cas de Apple, les aplicacions han de ser revisades per Microsoft abans de ser publicades a la botiga, així s'evita la publicació d'aplicacions fraudulentas o possibles aplicacions malicioses, augmentant el nivell de seguretat de la botiga de cara als usuaris finals.

2.2.3 BlackBerry

Com ja s'ha comentat, BlackBerry va ser un dels precursors dels Smartphones actuals, tot i que orientat a executius, ja disposava de característiques molt importants en els Smartphones

d'avui en dia, com ara la gestió del correu electrònic o la organització de l'agenda personal i el calendari. El sistema operatiu creat per RIM, disposa de multi-tasca fet que permet un ús intensiu de les diferents particularitats del sistema de forma simultània. Un dels punts importants d'aquest sistema, és que disposa una maquina virtual Java, que proporciona suport per aplicacions escrites en Java.

El sistema operatiu disposa d'una eina de sincronització amb l'ordinador d'escriptori, on instal·lant el programari "BlackBerry Desktop Software" permet crear còpies de seguretat de les dades, contactes, agenda etc. Per altra banda, amb aquesta eina es permet una sincronització amb els gestors personals de dades, com per exemple "MS Outlook". Finalment, cal destacar la sincronització a través de la xarxa sense fils, amb comptes de "MS Exchange Server", permetent l'intercanvi de dades de correu electrònic, calendari, contactes, etc.

Les versions del sistema operatiu han evolucionat constantment des dels seus inicis per adaptar-se a les necessitats sobretot del mercat empresarial. Al 1999 es comercialitzava la primera versió del sistema (RIM OS 1.0), que estava destinat als buscapersones del propi fabricant. Aquest sistema ja disposava de correu electrònic, agenda i calendari. Un any més tard, (2000) va aparèixer la segona versió, que ja s'adaptava més a les PDA, afegint la capacitat de navegació web i amb l'aparició dels primers jocs per la plataforma. Fins al 2002 però, no va aparèixer la primera versió per telèfons mòbils, "RIM OS 3.0", que entre altres novetats va aparèixer el suport per Java. Així, es va aplanar el camí per tal que al 2004 aparegués "BlackBerry OS 4.0", els primers dispositius més moderns, amb pantalla a color i l'aparició del xat de BlackBerry que va suposar un gran èxit. Aquesta versió va durar varis anys, evolucionant amb sub-versions fins al 2008, on es va llançar "BlackBerry OS 5.0". Aquest sistema va suposar la popularització de les BlackBerry en el públic general, pensat sobretot per entrar en el sector del mercat més jove, amb models més assequibles i l'avantatge de permetre accés a internet des de la xarxa mòbil. A partir del 2008 van aparèixer els primers terminals BlackBerry amb pantalla tàctil que utilitzaven aquesta versió del sistema. Fins al 2010 però no es va passar a la següent versió, "BlackBerry OS 6", que va aportar sobretot millores visuals i d'estabilitat del sistema. Un dels punts més avantguardistes aportat per aquest sistema va ser la integració de les xarxes socials amb notificacions pròpies del sistema. Al 2011 es va millorar la versió anterior, amb "BlackBerry OS 7", que tot i aportar poques millores substancials, aporta un aire més modern al sistema. Aquesta versió és la més utilitzada en l'actualitat, tot i que al 2013 va aparèixer "BlackBerry 10", juntament amb un nou dispositiu tàctil, més semblant amb la resta de Smartphones de la competència, en un intent de suplir les carències que aquest sistema estava mostrant respecte als seus competidors. De totes maneres, tot i la modernització de la companyia, sembla que es travessa un moment complicat a nivell econòmic i sobretot referent a la quota de mercat assolit per aquesta empresa. El temps i el mercat decidiran sobre la supervivència d'aquest sistema operatiu per a telèfons intel·ligents.

2.2.4 Android

El sistema operatiu Android és a dia d'avui el més utilitzat del planeta per a dispositius mòbils intel·ligents, el fet de ser un sistema de codi obert ha promocionat que moltes empreses fabricants de mòbils l'utilitzin per als seus dispositius. D'aquesta manera, es poden trobar dispositius de totes les gammes, obrint el mercat dels telèfons intel·ligents a tots els públics.

Aquest sistema, com ja s'ha dit no només s'utilitza per a mòbils, sinó que també en tauletes o inclús en televisions d'última generació (Smart TVs). Android està basat en un sistema Linux amb la idea inicial de adaptar-lo a dispositius mòbils amb pantalla tàctil.

La botiga d'aplicacions disponible per al sistema s'anomena "Play Store", (en els seus inicis "Market") i permet publicar aplicacions a qualsevol persona. L'únic requisit per publicar aplicacions és registrar-se com a programador i pagar una quota única de 25\$ (uns 20 euros al canvi). Per altra banda, i a diferència de la resta de botigues d'aplicacions, és que quan un programador publica una aplicació, aquesta està disponible a l'instant en la botiga d'aplicacions, és a dir, no passa per un control previ per part de Google, disminuint relativament la seguretat de les aplicacions que es poden trobar a la botiga. Es comenta en termes relatius, ja que és la comunitat d'usuaris qui té el poder per avisar d'aplicacions malicioses o fraudulentas, d'aquesta manera Google, en detectar queixes dels usuaris revisen les aplicacions denunciades i procedeixen adequadament en cada cas. Amb aquest sistema s'aconsegueix una agilitat en el mercat d'aplicacions que no es troba en els altres sistemes, per altra banda es promociona l'aparició de nous programadors. Així, al ser un sistema de codi obert, és la pròpia Google qui posa a disposició dels programadors tot un conjunt d'eines i informació per poder aprendre a desenvolupar aplicacions per Android, motivant encara més l'aparició de nous desenvolupadors. Així, el "Play Store" ja ha superat el milió d'aplicacions, de les quals dues terceres parts són gratuïtes.

De totes maneres, Android és majoritàriament de codi obert sota una llicència Apache. Les aplicacions són programades en Java, en un entorn orientat a objectes i s'executen amb un nucli de llibreries Java sobre una màquina virtual Java. Aquesta ha estat dissenyada específicament per adaptar els requeriments de les aplicacions al el rendiment de cada terminal en concret, aquesta màquina virtual s'anomena "Davitk Virtual Machine", que compila directament en temps d'execució. Tot i així, al tractar-se d'un nucli Linux, el sistema Android està programat amb llenguatge C++, tot i que el codi del sistema disposa parts en Java, i que la programació d'aplicacions es realitza en Java.

Historia

Tal com s'ha explicat, el sistema operatiu va ser desenvolupat per "Android Inc." al 2003, empresa que Google va comprar al 2005. D'aquesta manera, es va presentar el sistema al 2007 juntament amb la "Open Handset Alliance" (OHA), que com s'ha comentat anteriorment, és un consorci d'empreses de hardware, software i telecomunicacions amb la intenció d'avançar

amb els estàndards de sistemes oberts. Al 2008 es va comercialitzar el primer dispositiu amb Android que va ser fabricat per HTC. Al 2010, Android ja era el sistema operatiu mòbil més utilitzat del mercat, o dit d'una altra manera, existien més dispositius amb alguna versió de Android que de qualsevol altre sistema operatiu. Al 2011, a Estats units gaudia d'una quota de mercat del 43%, mentre que a nivell mundial la quota d'Android superava el 50% de dispositius.

Com ja s'ha comentat, es va iniciar una "guerra" (a nivell de mercat), entre Apple i Android, on ambdós pretenien dominar el major percentatge del mercat, però les estratègies comercials sempre han estat oposades entre ambdós competidors. Per una banda, s'ha vist que Apple només permet utilitzar iOS en els seus dispositius, i per altra banda Android és un sistema de codi obert. D'aquesta manera, la resta de fabricants de mòbils van optar per Android, fet que va suposar un increment massiu de terminals amb el sistema operatiu de Google llançats al mercat. Aquest fet també va propiciar que la comunitat de programadors per Android creixés de manera exponencial.

Versions

L'historial de versions del sistema operatiu va començar amb la seva presentació al 2007 amb una versió Beta. Un any més tard, en llençar al mercat el primer dispositiu Android es va inicialitzar amb la versió 1.0. El nom en clau de cada versió està relacionat amb el nom d'un postres en anglès, on augmenta la inicial en ordre alfabètic a cada versió. Tot seguit s'exposa el seguit d'actualitzacions rebudes pel sistema operatiu des dels seus inicis juntament amb el nom en clau de cada versió. Per norma general, cada actualització corregeix determinats errors o inestabilitats del sistema, alhora que aporta noves funcionalitats.

- 1.0 Apple Pie (pastís de poma), setembre de 2008
- 1.1 Banana Bread (pa de plàtan), febrer 2009
- 1.5 Cupcake (magdalena), abril 2009
- 1.6 Donut (bunyol), setembre 2009
- 2.0 Eclair (xuixo), octubre 2009, va tenir una actualització menor, a la versió 2.1 al gener de 2010 amb correcció d'errors.
- 2.2 Froyo (en anglès "Frozen Yogurt", iogurt gelat), maig 2010
- 2.3 Gingerbread (pa de gingebre) desembre 2010, va sofrir diverses actualitzacions per millorar l'estabilitat i corregir alguns errors, arribant fins a la versió 2.3.7 al setembre de 2011.

- 3.0 Honeycomb (bresca de mel), febrer de 2011 (només per tauletes, es va decidir separar el desenvolupament per Smartphones i tauletes per millorar-ne el disseny i la estabilitat per separat). Va sofrir diverses actualitzacions, arribant a la versió 3.2.6 al febrer de 2012.
- 4.0 Ice Cream Sandwich (gelat de nata amb galeta), octubre 2011. Amb aquesta versió es va reagrupar el desenvolupament per Smartphones i tauletes que s'havia separat amb la versió 3. Va rebre diverses actualitzacions menors, arribant fins a la 4.0.4 al novembre de 2012.
- 4.1 Jelly bean (caramels de gelatina) juny 2012. Aquesta versió ha rebut varies actualitzacions importants, i varies de menors. En cada actualització important Google anunciava que s'aportava un nou sabor de caramel de gelatina:
 - 4.2 Jelly Bean Gummy Bear (osset de goma) octubre 2012
 - 4.3 Jelly Bean Gominola (llaminadura) juliol 2013
- 4.4 Kit Kat (barres de xocolata comercialitzades per Nestlé), anunciat el setembre 2013, s'espera que aparegui al mercat a finals d'octubre 2013

Actualment, segons dades de Google del mes d'octubre de 2013, la versió més utilitzada de forma activa és Jelly Bean, les dades es publiquen en el portal web de per al desenvolupament d'Android i es poden observar tot seguit:

Versió	Nom en clau	Percentatge d'ús
2.2	Froyo	2.2%
2.3.3 - 2.3.7	Gingerbread	28.5%
3.2	Honeycomb	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	20.6%
4.1.x	Jelly Bean	36.5%
4.2.x		10.6%
4.3		1.5%

Figura 12. Taula de versions Android i percentatge d'ús. [6]

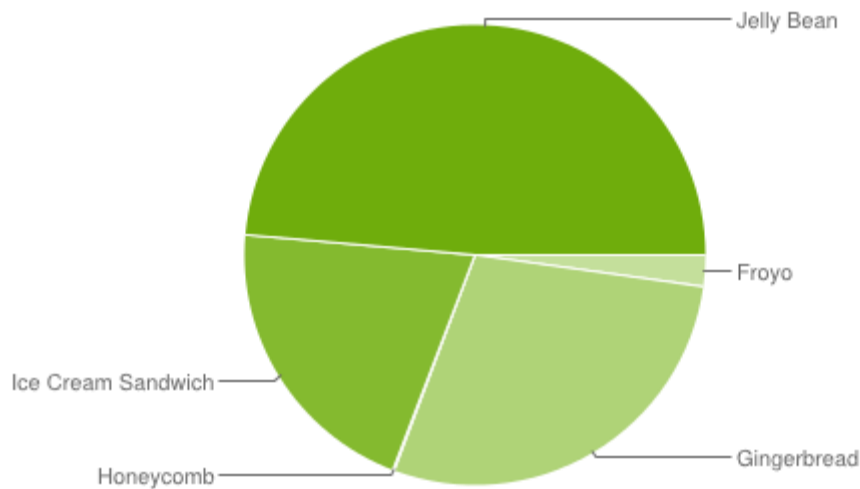


Figura 13. Gràfic de versions Android i percentatge d'ús. [6]

Motius per escollir-lo

En primer lloc, tal com s'ha comentat, Android disposa d'una comunitat d'usuaris i programadors molt gran. En ser de codi lliure el sistema, permet que moltes de les tasques a realitzar (a nivell de programació) ja quedin exemplificades i explicades directament de la mà de Google. Per altra banda, permet gaudir de la immensa comunitat de programadors, ja que davant d'un problema, molt sovint altres l'hagin sofert, i per tant es poden trobar solucions sense tenir que "reinventar la roda", alhora que permet cooperar amb la comunitat, aportant possibles solucions o problemes detectats durant el desenvolupament.

Per altra banda, s'ha vist que a nivell econòmic és molt més factible començar a programar per Android degut als reduïts costos inicials, és a dir, pagant 25\$ podem publicar una aplicació a la botiga sense cap problema. Si més endavant no volem seguir publicant aplicacions, o al cap de 3 anys volem publicar una nova aplicació ja no tindrem cap problema, amb aquest únic pagament ja obtenim una llicència vitalícia. Per altra banda, en els altres sistemes s'ha de pagar de forma anual una quota de aproximadament 70 – 75 euros, i en el cas de Apple, s'ha de disposar d'un ordinador amb "Mac OSX", mentre que amb Android es pot desenvolupar des de qualsevol ordinador.

Finalment, si programem per Android, tenim una possibilitat d'èxit superior davant les altres plataformes degut a la quota de mercat que disposa actualment Android. Per contra, si ens fixem en BlackBerry per exemple, podem intuir que la plataforma està sortint del mercat, amb una quota d'ús a la baixa, de manera que una aplicació per aquest sistema tindrà menys possibilitats de resultar un èxit, o si més no, d'arribar a un gran volum de públic.

Resumint:

- Codi lliure
 - o Guies de programació oficials
 - o Gran comunitat de desenvolupament
 - o Exemples de codi

- Econòmic
 - o Llicència vitalícia de 25\$
 - o No és necessària una inversió en equips informàtics específics
 - o Cursos i guies de programació gratuïts

- Arribar a un gran públic
 - o Quota de mercat en creixement
 - o Superior quota de mercat que la competència

2.3 Jocs en Smartphones

Com s'ha vist fins ara, els Smartphones han canviat completament la forma en que les persones utilitzen el telèfon mòbil, aportant una vessant orientada a l'oci que fins l'aparició del primer telèfon intel·ligent no tenia cabuda en aquest sector. Un dels apartats que han vist una oportunitat ha estat el dels vídeo-jocs. Fins aleshores, només és jugava en vídeo-consoles, connectats a una televisió o en consoles portàtils, amb unes prestacions reduïdes i jocs poc econòmics.

Així, amb l'aparició dels Smartphones i les seves botigues d'aplicacions, el sector del vídeo-joc ha sabut trobar un forat on expandir-se, trobant alternatives que anteriorment ni es plantejaven, com és el cas d'utilitzar les pantalles tàctils dels dispositius, aprofitar els sensors de moviment del dispositiu o la connectivitat a internet en mobilitat per posar-ne algun exemple. D'aquesta manera, han pogut desenvolupar noves formes d'entreteniment, ja sigui amb jocs senzills per emplenar els buits de temps sorgits en el dia a dia de la vida moderna (espera del autobús, etc.) No cal oblidar però, que els dispositius han evolucionat molt a nivell de prestacions tècniques, permetent el desenvolupament de jocs molt més elaborats a nivell gràfic, convertint-los inclús en les noves vídeo-consoles portàtils. Finalment, un dels punts molt important en aquest procés, ha estat la connexió a internet que disposen generalment aquests dispositius, fet que propicia l'aparició de jocs multi-jugador online, ja sigui síncrons, compartint un mateix entorn de joc, on cada jugador pot realitzar accions que tenen efecte en la pantalla de la resta de jugadors, com jocs asíncrons, on no és necessària la connexió simultània de ambdós jugadors, de manera que el joc es desenvolupa per torns, per tant, caldrà esperar que l'altre jugador trobi el moment per obrir el joc i realitzar la seva acció i viceversa.

2.3.1 Tipus de jocs més utilitzats

El mercat dels vídeo-jocs per Smartphones és molt ampli i s'hi poden trobar jocs de tots tipus. La diversitat és tant gran, que s'escapa de l'objectiu d'aquest document llistar totes les varietats disponibles, per tant es mostrarà els estils més comuns.

El primer estil, és un dels que gaudeix de més èxit tot i ser un joc molt senzill i repetitiu, provoca força addicció als jugadors. Aquest és denomina amb l'estil de puzzle, i es tracta del típic joc d'unir boles colors a l'estil 3 en ratlla o 4 en ratlla, però en un marc limitat. Aquest tipus de joc se'n poden trobar moltes variants, alguns amb un marc ple de boles aleatòries, on es permet intercanviar-ne dues de consecutives per formar unions de 3 boles del mateix color, en altres l'usuari ha de disparar una bola nova a cada torn, emplenant el marc de boles i aconseguint unir-ne per colors evitant emplenar tot el marc, etc. Les variants d'aquest estil són infinites, en uns es tracta de boles de colors, en altres caramels de colors, en altres són ossets, però en tots la finalitat és la mateixa, crear unions d'un mateix color/espècie de 3 o més unitats i fins l'infinit. Normalment, l'addicció es genera en permetre determinades fites que l'usuari haurà d'assolir, ja sigui en base a puntuacions, avançar en nivells més complicats o poder compartir en les xarxes socials els avenços aconseguits per tal de competir amb els amics.

Un altre estil molt recurrent, és el joc d'acció. Aquest estil engloba moltes variants, però generalment es sol conèixer amb el nom de jocs "arcade". En aquest tipus, l'usuari controla un personatge i ha d'avançar per un determinat espai. Generalment es pot trobar obstacles que haurà d'esquivar o enemics que podrà matar. Aquesta variant es pot dividir en molts gèneres, un d'ells és el clàssic arcade on un personatge avança per un món ple d'obstacles, amb accions possibles com saltar o disparar.

Un altre pot ser el "shoot'em up", on es tracta de controlar un personatge o un vehicle que avança de forma continua i sense parar en un món ple d'obstacles a evitar o disparar. En aquest estil també es pot trobar gèneres derivats, com seria el cas de controlar un personatge amb visió en primera persona, permetent els controls de direcció i de disparar, d'aquesta manera el jugador ha de moure's per un mapa ple d'enemics que generalment també desapareixen, en aquest cas igual que en la resta, l'objectiu és sobreviure el màxim de temps possible generant una puntuació el més elevada possible.

Un estil de jocs molt orientat als dispositius tàctils, són els jocs d'enginy i en aquesta vessant es pot trobar molta varietat. Es tracta de moure determinats objectes de una manera determinada per obtenir un resultat determinat. Generalment es basa en permetre un nombre limitat de moviments, incentivant el raonament estratègic, lògic o inclús de precisió del usuari.

Per altra banda, es poden trobar jocs de simulació, ja sigui d'esports com de carreres. Aquests generalment disposen d'un apartat gràfic molt treballat i solen consumir molts recursos en el

dispositiu, per tant es tracta d'un estil de joc no tant orientat a la mobilitat, ja que els usuaris poden quedar-se sense bateria molt ràpidament, a més generalment les partides solen ser llargues. Tot i així, aquest estil gaudeix de força èxit sobretot en els dispositius més usats en les llars com són les tauletes.

Un dels estils amb una popularitat creixent són els jocs d'estratègia, ja sigui per torns o en temps real. Aquests sovint permeten partides multi-jugador i es solen basar en jocs tradicionals de taulell, o inclús es poden trobar jocs de rol.

2.3.2 Exemples de jocs

En aquest apartat es mostrarà algun exemple dels estils de joc més descarregats en les botigues d'aplicacions.

Puzle

“Candy Crush Saga”, un “top vendes”, el més descarregat en aquest estil en l'actualitat (entre 100 milions i 500 milions de descàrregues a la botiga de Android).



Figura 14. Imatges de Jocs – Candy Crush Saga.

Arcade

“Ralph’s world”, una adaptació del clàssic “Super Mario World” (entre 1 milió i 5 milions de descàrregues a la botiga de Android).



Figura 15. Imatges de Jocs – Ralph’s World.

Shoot'em Up

“JetPack JoyRide”, shoot'em up senzill amb missions que generen una gran addicció tot i ser molt repetitiu. (entre 50 milions i 100 milions de descàrregues a la botiga de Android).

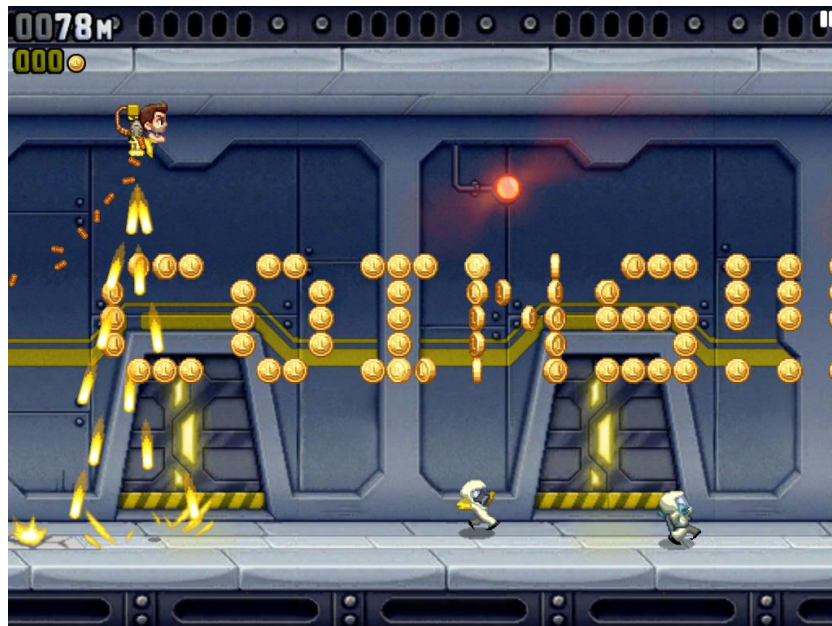


Figura 16. Imatges de Jocs – Jetpack Joyride.

Enginy

“Angry Birds”, s’ha convertit en tot un clàssic, es basa en disparar un nombre determinat d’ocells per eliminar tots els porcs de cada nivell. Tot i que aquest joc es podria englobar en alguna altra categoria com “arcade”, cal tenir present que sovint serà necessari aplicar una bona estratègia i disposar d’un bon enginy per poder superar els nivells proposats. (entre 100 milions i 500 milions de descàrregues a la botiga de Android, tenint en compte només la versió bàsica ja que n’ha sorgit diverses variants amb diferents ambientacions)

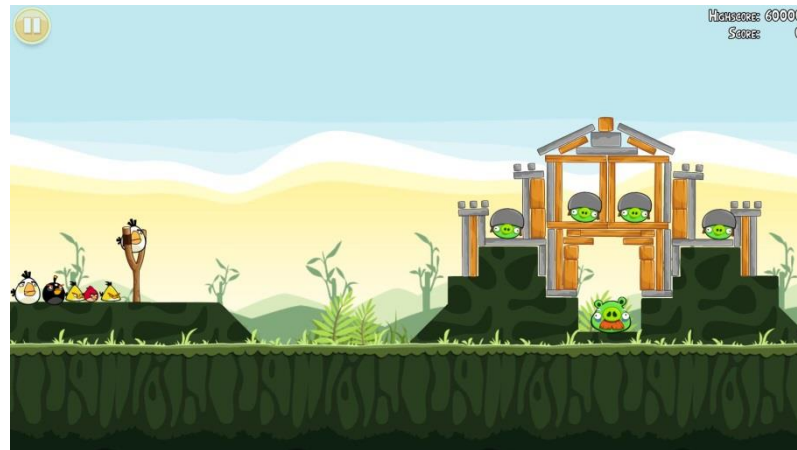


Figura 17. Imatges de Jocs – Angry Birds.

Val a dir que en aquest punt només s’ha mostrat jocs gratuïts, d’aquesta manera es pretén mostrar l’èxit que es pot aconseguir permetent la lliure descarrega. De totes maneres, el fet que el joc sigui gratuït no implica que els programadors no en puguin treure profit econòmic, ja que existeixen moltes maneres de obtenir ingressos amb les aplicacions sense cobrar per descàrregues, com per exemple mitjançant la publicitat o els pagaments dins de la aplicació (per obtenir determinades millores o per avançar més ràpid en el joc superant nivells complicats per exemple).

2.4 Programar per Android

Amb Android es programa amb llenguatge Java, mitjançant una adaptació d’aquest, ja que no conté absolutament totes les classes disponibles en Java però sí una gran majoria. Per altra banda, es tracta d’una programació orientada a objectes, on es permet l’ús de bases de dades relacionals mitjançant el llenguatge SQLite, que ve integrat amb el sistema Android.

2.4.1 Sistema de desenvolupament

En aquest apartat es procura presentar breument el sistema utilitzat per desenvolupar aplicacions per Android per tal d'obtenir una idea general de la metodologia utilitzada.

En primer lloc és necessari un entorn de desenvolupament, generalment s'utilitza "Eclipse", (però es pot utilitzar qualsevol altre IDE), tot i que recentment Google ha creat el seu propi entorn anomenat "Android Studio". De totes maneres, l'entorn estàndard actualment segueix sent "Eclipse", ja que "Android Studio" encara es troba en fase beta.

En segon lloc, cal disposar del que s'anomena "SDK Android", que és la base del sistema operatiu amb totes les versions disponibles fins al moment. Paral·lelament serà necessari obtenir el "ADT" (Android Development Tools), que és una extensió per l'entorn de programació (Eclipse), per tal d'incloure-hi les funcionalitats necessàries per compilar aplicacions per Android. ADT resulta un pont entre Eclipse i l'SDK d'Android.

D'aquesta manera, al crear un nou projecte Android, l'entorn de programació ja crea la estructura de fitxers necessaris per al funcionament d'una aplicació per Android. En aquest punt, és important tenir clar com s'estructura una aplicació.

Primerament es disposa del codi java pròpiament dit, que es crea dins el directori "src". Aquí s'hi poden crear tants fitxers i classes java com siguin necessaris per al desenvolupament en concret.

Seguidament, és important mencionar que l'apartat visual de les aplicacions es separa del codi java. Així, la interfície gràfica de cada "pantalla" (en Android es parla de "activitats"), es defineix en un fitxer XML. Aquests fitxers es troben al directori "layout". Per altra banda, com que Android pot suportar múltiples dispositius amb configuracions físiques diferents (pantalles de totes mides i resolucions), un mateix "layout" no es visualitza de la mateixa manera en una pantalla de 3 polzades i una resolució de 320x480 píxels que en una pantalla de 5 polzades i 1920x1080 píxels, per tant, es poden crear variacions de un mateix fitxer XML per a cada conjunt de mides i resolucions. També és important decidir com es visualitzarà una aplicació quan l'usuari giri 90º el dispositiu, és a dir, posant el dispositiu en horitzontal. Així, es pot crear el directori "layout-land", on s'emmagatzema el layout corresponent a la visualització en horitzontal, o per exemple "layout-large" on s'emmagatzemaria el layout corresponent a pantalles entre 4 i 7 polzades. Queda doncs en ma de cada programador adaptar o no les visualitzacions de la seva aplicació a diferents tipus de dispositius i orientacions.

Finalment, pot resultar interessant afegir imatges, vídeos o sons com a recursos de les aplicacions desenvolupades. Per això es disposa de varis directoris. En primer lloc es troba el directori "drawable" que permet emmagatzemar imatges, aquest, al igual que amb els layouts permet alternatives del mateix estil, per exemple "drawable-hdpi", on s'emmagatzemarien les

imatges a utilitzar per als dispositius de alta densitat de píxels per polzada (la densitat dels píxels és una unitat de mesura que relaciona la mida de les pantalles amb la resolució d'aquestes). Així, el sistema accedirà al recurs corresponent de forma automàtica en funció de les característiques del dispositiu on s'executi l'aplicació. L'objectiu d'aquesta separació, és que una mateixa aplicació es visualitzi igual o de la forma més semblant en tots els dispositius possibles, ja que per exemple, una imatge amb una resolució de 240x240 en un dispositiu amb una pantalla de 320x480 píxels de resolució, ocuparà gairebé tota la pantalla, mentre que en un dispositiu de 1920x1080 píxels suposarà menys d'una quarta part. Per altra banda, es troba el directori "raw" que és on s'emmagatzemen els fitxers de so i vídeo.

Al moment de crear el codi, el programador pot accedir a les estructures creades en els apartats visuals (layouts), com per exemple a un botó per programar-ne un funcionament. Així, per vincular cada un d'aquests elements dins de cada layout amb l'apartat de codi, es realitza gràcies a un fitxer de recursos que genera automàticament el propi entorn de programació. D'aquesta manera, el programador no s'ha de preocupar de on es trobarà el recurs en concret en el moment d'execució ni de si haurà d'agafar el recurs dins del layout horitzontal ni de quina resolució utilitzi el dispositiu, simplement haurà d'especificar el nom del element en concret (per exemple: `button1`) precedit de "R.id", en aquest cas doncs quedaria de la següent manera: "R.id.button1". Així el sistema ja s'ocuparà de triar el recurs corresponent en cada cas. El mateix succeirà amb els recursos d'imatges, sons o vídeos, on simplement s'haurà d'especificar el directori on es trobi el recurs precedit de "R." (per exemple, una imatge emmagatzemada a `drawable-hdpi` amb el nom de "imatge.png", s'hi pot accedir amb "R.drawable.imatge" i el sistema agafarà en cada cas la imatge de la resolució corresponent en cada dispositiu on s'executi l'aplicació.

Per altra banda, es poden incloure llibreries desenvolupades per terceres persones, per tal de simplificar determinades tasques comunes, ja sigui per realitzar determinats càlculs, o directament per implementar determinades estructures gràfiques que requereixin una programació amb moltes classes i molt escabrosa. Per aquesta finalitat es disposa d'un directori anomenat `lib`, que serà on es permetrà incloure llibreries externes o frameworks determinats.

2.4.2 Desenvolupament de jocs: Frameworks gràfics

El propi sistema Android disposa un framework que permet desenvolupar mitjançant la API gràfica OpenGL ES 1.0 i OpenGL ES 2.0. De totes maneres, pot resultar complicat desenvolupar jocs mitjançant les eines proporcionades per Android, ja que es baixa a un nivell purament gràfic (directament instruccions OpenGL), per tant, pot resultar complicat centrar-se únicament amb el desenvolupament pròpiament dit ja que molts aspectes s'hauran de configurar i programar per separat, abstraient el programador del que és la programació del joc. Per aquesta raó, han aparegut múltiples Frameworks orientats al desenvolupament de jocs

per Android, que faciliten moltes tasques comunes en el desenvolupament de la gran majoria de jocs, com podria ser per exemple el motor gràfic o la física del joc. En aquest apartat es resumeixen breument els més utilitzats per tal d'obtenir una visió general de cada un d'aquests i permetre més endavant elegir quin serà el que s'utilitzarà en el desenvolupament d'aquest projecte.

AndEngine

Es basa en mòduls, de manera que permet diverses extensions per adaptar-se a cada desenvolupament en concret. Utilitza el motor de física "Box2D", i permet utilitzar protocols de xarxa tal d'implementar arquitectures del tipus client-servidor i desenvolupar així jocs multi-jugador.

Es programa amb llenguatge Java però mitjançant instruccions pròpies del framework, de totes maneres la corba d'aprenentatge és molt baixa, de manera que no cal adquirir grans coneixements per poder-lo utilitzar. Cal mencionar que no disposa d'una documentació molt extensa, però per contra es poden trobar múltiples exemples didàctics.

Per altra banda, a nivell gràfic es treballa mitjançant el que s'anomena "Sprites", que són la representació gràfica de cada objecte o element del joc. Aquests poden ser programats a nivell de codi per tal d'interactuar amb la resta d'elements, permetent la detecció de col·lisions per exemple.

Finalment, cal destacar que el framework només permet desenvolupament per Android, sense tenir en compte una possible portabilitat del joc a altres plataformes com web, escriptori o iOS.

LibGDX

La principal característica d'aquest framework és que permet desenvolupar jocs multi-plataforma mitjançant un únic codi java. D'aquesta manera es pot exportar els projectes ja sigui per Android, escriptori o html.

Per altra banda, la corba d'aprenentatge és més elevada que l'anterior framework, però per contra es disposa d'una documentació molt extensa tant de forma oficial com per part de tercers, ja que es poden obtenir múltiples guies i vídeo-guies realitzades per tercers, (s'ha trobat multitud de blogs externs especialitzats amb aquest framework). Per tant, disposa d'una comunitat suficientment àmplia per obtenir suport en cas d'errors, a banda de múltiples codis d'exemple per tal d'aprendre'n el funcionament.

A nivell gràfic, resulta molt potent, ja que treballa directament sobre OpenGL 2.0, i possiblement aquesta sigui la causa de la dificultat inicial en l'aprenentatge. De totes maneres, cal destacar que a nivell de rendiment, aquest framework obté millors resultats que qualsevol altre, sobretot en carregues gràfiques molt importants, inclús millor que programant amb el propi framework d'Android.

Finalment, és important mencionar que no disposa característiques pròpies per realitzar tasques de xarxa i per tant pot resultar complicat desenvolupar jocs multi-jugador. De totes maneres, aquesta funcionalitat es pot implementar directament gràcies a les funcionalitats pròpies de Java. Per altra banda, existeixen extensions que permeten implementar aquestes funcionalitats, com per exemple l'extensió per la plataforma "Google Play Game Services", que tot i ser molt recent i no estar gaire documentada, pot aportar varies millores pel que fa al desenvolupament multi-jugador.

Aquesta plataforma (Google Play Game Services - GPGS) permet utilitzar els servidors de Google per la implementació de la part servidor dels jocs desenvolupats per Android. Així, es poden abaratir costos de manteniment, a banda de permetre altres funcionalitats com compartir les puntuacions entre els amics de l'usuari a la xarxa social de Google (Google +), o permetre una classificació a nivell mundial o a nivell d'amistats de "Google +". Una altra funcionalitat interessant de GPGS són les fites o missions aconseguides per un usuari, d'aquesta manera es podria implementar un apartat de missions a realitzar, així, les actualitzacions de missions es realitzarien directament al costat servidor, evitant actualitzar constantment el joc una vegada finalitzat.

3. Disseny i implementació

3.1 Estructura del joc

En primer lloc, es detallarà la fase de disseny i implementació del joc, començant per exposar l'esquelet de l'aplicació i repassant totes les funcionalitats d'aquesta. Per altra banda, s'explicarà com i perquè s'ha pres determinades decisions en l'etapa d'implementació, tractant així d'exposar i clarificar el funcionament intern del joc, tant a nivell de codi com de lògica estructural. Així doncs, es començarà analitzant l'esquelet a través d'un diagrama de flux.

Per entendre millor el funcionament, en aquest apartat es separa el disseny en dos apartats, la base del joc i les funcionalitats extra.

3.1.1 Base del joc

- Partida Simple
- Puntuacions
- Botiga de bons
- Configuració
- Informació

Partida Simple

Carrega tots els components visuals en memòria, arrenca el motor gràfic i comença a funcionar el joc.

Es crea un escoltador d'esdeveniments per poder atendre a pulsacions sobre la pantalla i poder determinar la voluntat del jugador.

Per altra banda, es crea un escoltador de col·lisions, de manera que quan el personatge topa amb algun objecte es desenvolupi la resposta corresponent.

Cal inicialitzar un comptador de "frames" reproduïts, que enregistri els metres virtuals avançats durant el joc en curs, a mode de comptabilitzar la puntuació de la partida.

Finalment, quan el personatge mor en una col·lisió, s'ha de registrar la puntuació obtinguda a la base de dades de puntuacions i tornar a la pantalla inicial.

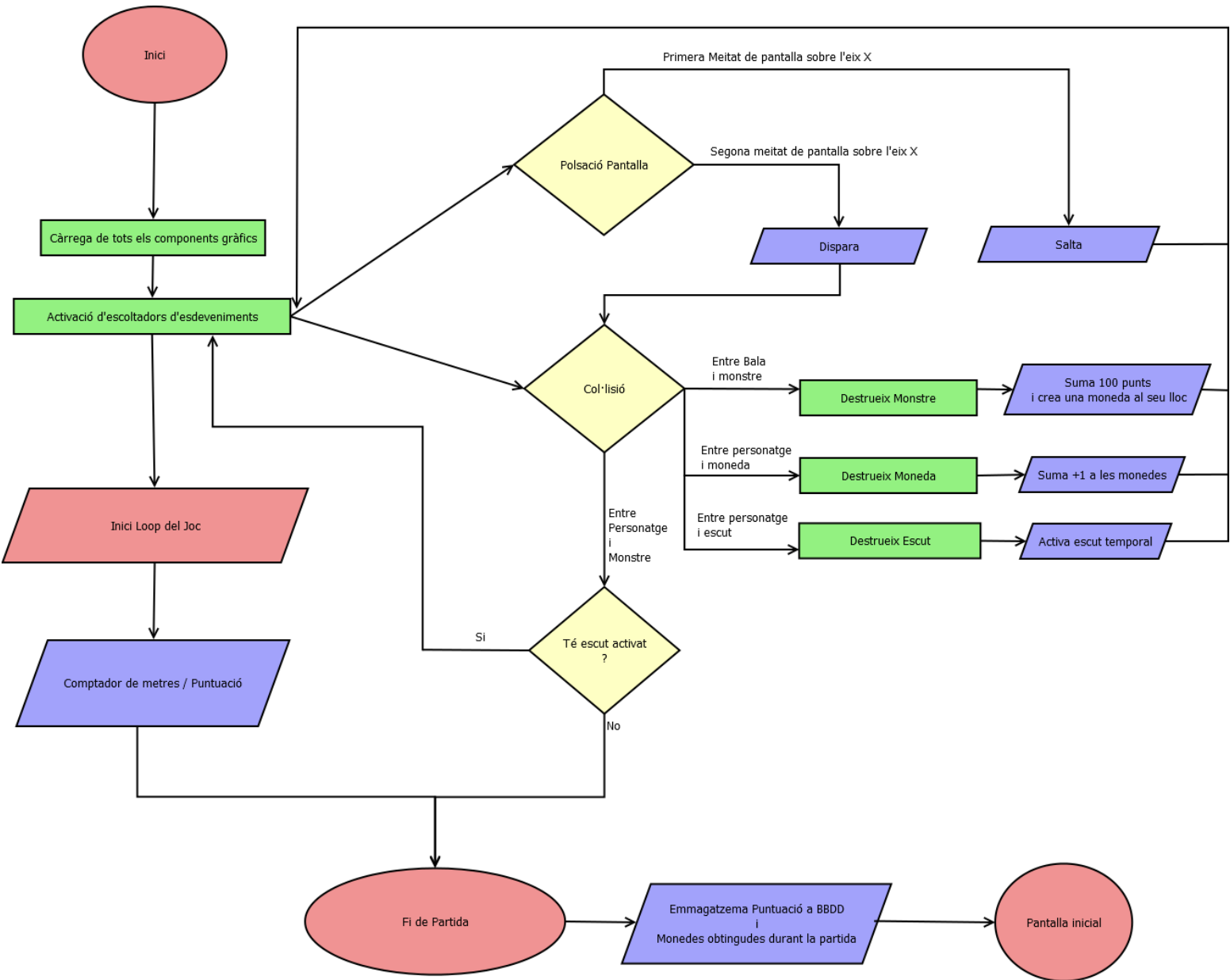


Figura 18. Diagrama de Flux – Partida simple.

Puntuacions

Les puntuacions es basen en un llistat, ordenat de major a menor de les 10 millors puntuacions obtingudes, cadascuna acompanyada amb el nom del jugador i la data de la partida. Les dades de les puntuacions es treuen d'una base de dades relacional, que conté per una banda el llistat de puntuacions juntament amb la hora i l'usuari i per altra banda el llistat d'usuaris. Per entendre la base de dades es mostra un diagrama de blocs senzill per representar l'estructura de les taules i la relació establerta.

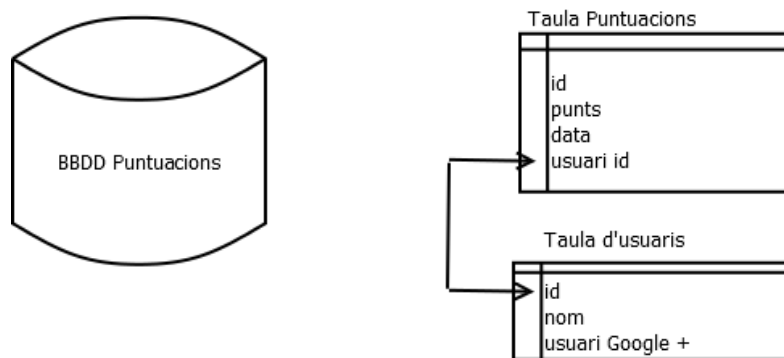


Figura 19. Diagrama de blocs – Base de dades de puntuacions.

Botiga de bons

La botiga de bons es tracta d'un llistat de possibles millores que pot adquirir l'usuari per al seu personatge, de manera que es pugui millorar la usabilitat del personatge durant les partides per tal d'obtenir millors puntuacions. Així, es basa en un llistat on apareix la funcionalitat addicional, el preu i un botó per adquirir-la. En cas que l'usuari no disposi de suficients monedes apareixerà un missatge informatiu, en cas afirmatiu es descomptarà el preu del total de monedes adquirides i s'aplicarà la bonificació de forma permanent.

Configuració

Al menú de configuració es permet triar si es vol reproduir musica de fons o no i es permet introduir el nom del usuari per tal que aquest quedi registrat amb les puntuacions. D'aquesta manera més d'un usuari podrà competir des d'un mateix dispositiu enregistrant puntuacions per separat.

Informació

Finalment, en la funcionalitat de informació, es mostra la descripció de la aplicació, s'explica les bases del joc i els controls.

3.2 Funcions extra: Google +

3.2.1 Descripció Google Play Games Services (GPGS)

Per poder utilitzar les funcionalitats extra de Google+, cal haver importat prèviament les llibreries de Google Play Game Services, fet que s'explica més endavant en aquest document.

Aquests serveis de Google permeten iniciar sessió amb l'usuari de la xarxa social de Google (Google+) i registrar tota l'activitat relacionada amb el joc per a cada usuari de la xarxa social. D'aquesta manera, es pot registrar les puntuacions dels usuaris de forma permanent als servidors de Google i comparar-les amb les de la resta d'usuaris, inclús es permet comparar-les entre les amistats de l'usuari o entre tot el món. Per tant, es permet una interacció entre amics i/o usuaris del joc creant una espècie de competició a nivell mundial o restringida només entre el cercle d'amistats de cada usuari.

Per altra banda, els serveis de Google permeten crear un registre de fites, de manera que cada usuari pugui obtenir unes medalles com a reconeixement o mèrit per seguir avançant en el joc, de manera que quan s'assoleix determinats objectius es desbloquegen les fites corresponents. Aquesta espècie d'incentius, a banda d'afavorir la fama de seguir jugant per part dels jugadors, permet mantenir els progressos dels usuaris de forma permanent, ja que un mateix usuari pot jugar des de diferents dispositius, o pot des-instal·lar el joc del seu dispositiu i reinstal·lar-lo al cap d'un temps, recuperant les dades emmagatzemades dins els serveis de Google, mantenint així tant els registres de puntuacions obtingudes anteriorment, com les fites aconseguides anteriorment, de manera que es podrà continuar el joc en el punt on es va deixar.

Finalment, els serveis de Google (Google Play Games Services) permeten realitzar partides multi-jugador, de manera que una vegada més, Google posa els seus servidors a disposició dels seus usuaris de la xarxa social, per tal de poder-los utilitzar com a punt d'enllaç entre usuaris i com a seu per poder realitzar partides multi-jugador.

3.2.2 Aplicació de GPGS al joc desenvolupat

- Inici/fi sessió
- Rànquing de puntuacions a Google +
- Rànquing de monedes a Google +
- Fites
- Partida multi-jugador

Inici/fi sessió

A la pantalla principal del joc, a banda de les funcionalitats bàsiques exposades en l'apartat anterior, es disposarà d'un botó per iniciar sessió a Google+, d'aquesta manera, si l'usuari disposa de més d'un compte associat al dispositiu se li demanarà amb quin dels seus comptes de Gmail desitja iniciar sessió. En cas de no disposar de cap compte l'usuari no podrà utilitzar la resta de funcionalitats extra proposades, ja que aquestes s'activen al iniciar sessió.

Cal tenir en comte que cada vegada que es canvia d'activitat (Activity = cada "pantalla" dins d'una aplicació Android), si s'ha iniciat sessió amb el compte de Google+ per utilitzar els serveis de Google (GPGS), es perd la sessió a cada canvi d'activitat i per tant s'ha de preveure de tornar a connectar de nou (sense que l'usuari hagi de intervenir-hi resultant transparent per aquest).

Així, a cada inici de sessió des de la pantalla principal de l'aplicació (des de l'activitat principal) s'haurà de comprovar si s'acaba de realitzar una partida o no i en cas afirmatiu s'haurà de preveure d'enviar les puntuacions a través de GPGS. Igualment, s'haurà de comprovar si es compleix alguna de les fites proposades i en cas afirmatiu enviar-la també a través de GPGS, també pot donar-se el cas que hi hagi alguna fita incremental, en tal cas caldrà actualitzar-ne l'estat, però sempre comprovant si s'acaba de jugar una partida instants anteriors a l'inici de sessió.

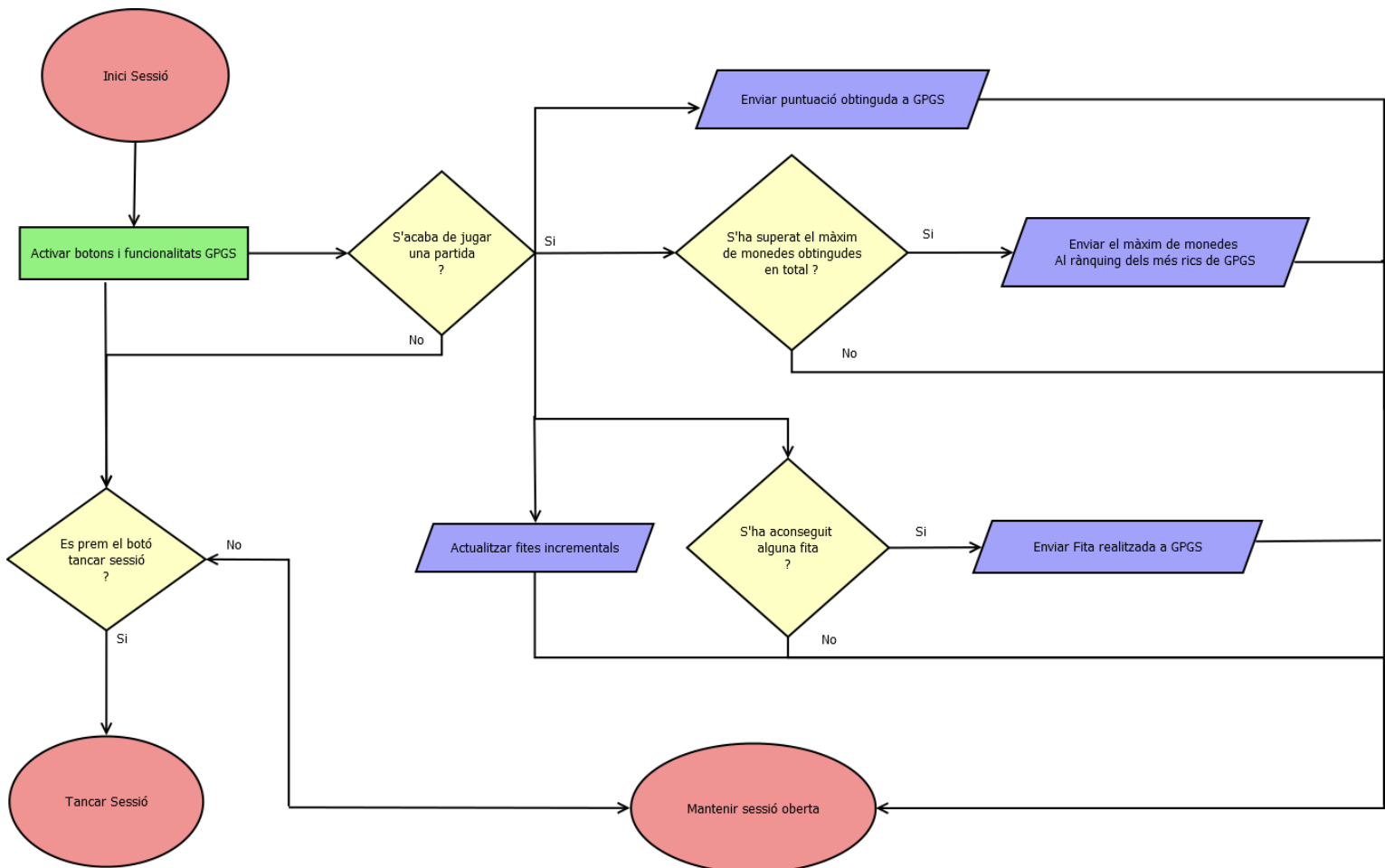


Figura 20. Diagrama de Flux – inici/fi sessió Google +.

Rànquings i Fites

Com s'ha vist, cada vegada que es finalitza una partida i es torna a l'activitat principal, si a aquesta anteriorment ja s'hi havia iniciat sessió amb el compte de Google+, es re-obrirà la sessió de forma automàtica i transparent per l'usuari (que no sabrà que realment s'ha desconnectat de Google+ durant la partida). Així, una vegada establerta la connexió amb els servidors de Google novament, s'envia al rànquing de puntuacions el resultat final obtinguda durant la darrera partida. Per altra banda, es suma les monedes obtingudes durant la darrera actuació amb el total aconseguit per l'usuari fins al moment, d'aquesta manera s'envia el nombre total de monedes actuals al rànquing de riquesa de GPGS.

El mateix es fa per les fites, en finalitzar cada partida i de la mateixa manera que el cas dels rànquings, una vegada re-establerta la connexió amb GPGS, es realitza les comprovacions per cada fita establerta (per exemple: "Jugar 15 partides"). Per tant, en cas que alguna fita s'hagi complert s'envia el missatge per desbloquejar-la a GPGS i obtenir així "la medalla".

Per al cas que s'ha posat d'exemple, al resultar una fita incremental no és necessari realitzar cap comprovació a través de codi, en aquests casos només cal enviar l'increment realitzat a GPGS, on prèviament s'ha configurat des de la consola de desenvolupament del compte del programador, cada fita a aconseguir i en cas que n'hi hagi d'incrementals ja s'hi ha establert el nombre total a obtenir.

3.3 Multi-jugador

L'apartat multi-jugador resulta més complex que la resta de funcionalitats que ofereix GPGS, ja que s'ha de definir una lògica estructural molt més important. Per aquesta raó, es separa de tot el que s'ha vist fins aquest punt. Com s'ha vist en els apartats anteriors, en establir sessió amb el compte de Google+ s'activa les funcionalitats de GPGS des de la activitat principal, però en cap moment s'ha parlat fins ara de multi-jugador. El que es farà doncs serà gestionar aquest apartat des d'una altra activitat de l'aplicació, de manera que una vegada l'usuari iniciï sessió amb el seu compte de Google+ des de l'activitat inicial, s'habilitarà (a banda de tot l'esmentat anteriorment), un botó per accedir a l'activitat de multi-jugador.

Tal com s'ha comentat, en canviar d'activitats la sessió activa de Google+ de l'usuari es perd, i per tant s'ha de tenir present que s'ha de re-connectar una vegada canviada l'activitat, a ser possible de forma transparent per l'usuari. De totes maneres es pot donar el cas que hi hagi alguna fallada o que l'usuari sofreixi una interrupció dels elements de xarxa durant el canvi d'activitat, per tant s'ha de preveure aquesta situació i permetre tornar a iniciar sessió de forma manual des d'aquesta nova activitat.

D'aquesta manera, des de l'activitat multi-jugador, s'habilita un botó per permetre tant iniciar sessió en cas que aquesta es perdi, com finalitzar sessió en cas que aquesta estigui activa i així

ho desitgi l'usuari. Així doncs, una vegada arribat a l'activitat multi-jugador i iniciada correctament la sessió, ja es pot plantejar quines funcionalitats permet GPGS per l'apartat multi-jugador, com es gestionarà i que es mostrarà en aquesta activitat.

- Partida simple
- Partida rapida multi-jugador
- Invitar amics
- Veure invitacions
- Tancar sessió

En cas de tancar la sessió per la raó que sigui, es manté l'activitat present, però només es mostra l'opció de partida simple i la de tornar a iniciar sessió.

Partida Simple

Com bé s'ha exposat, totes les funcionalitats de multi-jugador es gestionen exclusivament des d'aquesta activitat, i com és obvi, per poder realitzar una partida en mode multi-jugador cal trobar primerament un segon usuari disposat a jugar a través de la xarxa. Per aquesta raó, es pot donar el cas que es vulgui realitzar una partida multi-jugador i en aquell instant precis no hi hagi ningú disposat/disponible per dur a terme la partida. En aquests casos l'usuari s'hauria d'esperar de forma indeterminada fins que algú accepti la seva invitació o fins que un altre usuari també desitgi realitzar una partida multi-jugador.

Per evitar aquesta problemàtica (espera indefinida), des de l'activitat multi-jugador es permet realitzar partides simples, en mode local i de la mateixa forma que si s'iniciés partida des de l'activitat principal, amb la diferència que es manté la connexió amb GPGS i en el cas que arribi alguna invitació a una partida multi-jugador aquesta podrà ser atesa.

Però si s'analitza tot el que s'ha comentat fins al moment es pot apreciar una incongruència, si s'inicia una partida simple des de l'activitat multi-jugador, es canviarà d'activitat i per tant es perdrà la connexió amb GPGS i teòricament (segons el que s'ha comentat fins ara) no es podrà acceptar noves invitacions multi-jugador, aleshores, perquè no es permet aquesta funcionalitat al iniciar una partida simple des de l'activitat principal?

La justificació rau en el tipus de vista (View) que s'instancia des d'una banda i des de l'altra. Com s'ha exposat en aquesta documentació, des de l'activitat inicial, al executar una partida simple es canvia a l'activitat de partida, que està formada per una vista personalitzada (Custom View) que gestiona la partida. Per contra, en aquest cas l'apartat multi-jugador no és una activitat normal, sinó que és el que s'anomena en termes de programació d'Android una "Fragment Activity".

La peculiaritat d'una "Fragment Activity" (activitat fragmentada), és que permet disposar i canviar segons les necessitats entre diferents vistes mantenint l'entorn de l'activitat intacte, és a dir, no és necessari canviar d'activitat per representar diferents continguts a la pantalla. D'aquesta manera es disposa diverses vistes superposades com si es tractessin de capes, i en funció dels requeriments de l'aplicació en cada instant es pot ocultar o mostrar la capa/vista desitjada.

Aleshores, perquè no s'ha habilitat l'activitat principal en mode "Fragment Activity"? Les raons són dues, en primer lloc per evitar tenir que re-crear manualment l'activitat cada vegada que es desitgi re-establir l'entorn a zero, cada vegada que es realitza una partida per exemple, en aquest cas s'hauria de recrear l'entorn de l'activitat per buidar les variables o s'hauria de tenir en comte totes les casuístiques a tractar de forma manual i per tant es decideix aquesta alternativa per evitar una lògica estructural massa recargolada.

En segon lloc, cal recordar que el desenvolupament d'aquest joc procura servir no només per aplicar els conceptes adquirits durant la carrera sinó d'adquirir nous coneixements i habilitats, per tant, per tal de mantenir un codi ben estructurat amb els diferents apartats ben diferenciats i per poder així entendre millor tots els processos implicats es decideix separar el funcionament del mode multi-jugador d'aquesta forma.

Així doncs, reprenent l'explicació d'aquest apartat, en iniciar una partida simple des del mode multi-jugador, el que es fa és canviar el fragment que s'està mostrant (menú del mode multi-jugador) per al fragment corresponent a una instància del joc. D'aquesta manera l'usuari podrà amenitzar la espera jugant, però mantenint oberta la sessió de Google+ per poder paral·lelament escoltar invitacions a noves partides. En cas de ser convidat a una partida per algun amic dels seus cercles de la xarxa social de Google mentre aquest està a la espera realitzant una partida simple, es podrà recuperar la invitació una vegada finalitzada la partida des de l'apartat corresponent a visualitzar les invitacions rebudes.

Partida multi-jugador

Hi ha diverses maneres de començar una partida multi-jugador o millor dit, de trobar un oponent. Tal com s'ha exposat, l'activitat multi-jugador es basa en una activitat fragmentada (Fragment Activity), per tant, una vegada l'usuari accedeix a aquesta activitat es proposa les diferents opcions disponibles, la primera de les quals la hem definit a l'apartat anterior, però cal recordar que l'objectiu d'aquesta activitat és jugar simultàniament amb un altre jugador a través de la xarxa, per tant en aquest apartat s'exposa les diferents opcions disponibles per iniciar una partida multi-jugador.

El primer pas doncs és trobar un oponent disponible, i aquesta tasca es pot realitzar de diverses maneres. Però per poder convidar un oponent a una partida, cal crear prèviament una instància comuna que permeti intercanviar dades entre els usuaris implicats, aquesta instància

es representa o simbolitza com una sala (Room). Així doncs, per poder convidar a algú a jugar, en primer lloc s'ha de crear una sala comuna per la connexió o l'intercanvi de dades. De la mateixa manera, si rebem una invitació d'un altre usuari aquest haurà creat prèviament una sala de joc compartida.

Auto-Matching

Així doncs, una vegada creada la sala, en segon lloc es pot convidar usuaris a accedir a la sala i participar en la partida. Primerament s'exposa la funcionalitat de realitzar una cerca automàtica d'adversaris/jugadors disponibles, fet que en termes propis de GPGS s'anomena "Auto-matching". Aquesta cerca s'inicia al seleccionar la opció de realitzar una partida ràpida en mode multi-jugador i en tal cas cal canviar el fragment visible de l'activitat per al fragment "auto-matching" de GPGS. Cal tenir present que Google ja posa a disposició dels programadors el fragment corresponent a aquesta funcionalitat, però aquest punt s'explica més endavant en aquesta documentació.

Així, al iniciar la cerca automàtica d'oponents, són directament els serveis de Google que s'encarreguen de tota la tasca, comprovant quins usuaris hi ha disponibles en aquest instant que estiguin en aquest mode (cerca automàtica), per enviar una invitació de forma aleatòria a qualsevol d'aquests usuaris. D'aquesta manera GPGS enllaça de forma automàtica a dos usuaris que desitgin realitzar una partida multi-jugador amb un oponent aleatori, sense que aquest hagi de pertànyer a cap dels seus cercles d'amistats de la xarxa social. Una vegada establerta la connexió, s'uneix els dos usuaris a les sales compartides respectives, i es pot començar la partida en mode multi-jugador de forma simultània per als dos oponents, fet que se'n detalla el funcionament en els pròxims apartats.

Enviar invitacions

Com ja s'ha comentat, una altra manera de trobar un oponent és enviar una invitació a una amistat de l'usuari pertanyent a algun dels cercles d'amics de la xarxa social de Google. En triar aquesta opció, cal canviar el fragment visible en l'activitat multi-jugador, per fer visible la capa corresponent a enviar invitacions. En aquest cas, igual que en el mode "auto-match", Google ja disposa del fragment corresponent per tal que els programadors ho puguin utilitzar en els seus jocs, d'aquesta manera s'aconsegueix una estandardització d'aquest procés per a totes les aplicacions que disposin d'aquesta funcionalitat.

De totes maneres, si es desitja es pot crear un fragment propi per realitzar aquesta funcionalitat de forma personalitzada, tot i així en aquest projecte s'aprofitarà la feina feta per Google, ja que no es pretén re-inventar la roda sinó entendre i exposar el funcionament del procés per iniciar una partida en mode multi-jugador utilitzant els serveis de Google.

Així doncs, al accedir a la funcionalitat de convidar amics a una partida, es mostra una visualització en format de llista que permet escollir a quines amistats es desitja enviar una invitació. Una vegada enviada, es roman a la espera que algú la accepti per poder començar la partida. En aquest cas el primer que accepti la invitació serà admès a la sala creada per l'usuari i s'iniciarà doncs el procés d'inici de la partida (que com ja s'ha dit, s'exposa en un altre apartat d'aquesta documentació).

Per altra banda, es pot donar el cas que ningú accepti la invitació (ja sigui perquè no la veu o perquè aquesta és rebutjada). En cas que cap dels convidats respongui a la invitació, el procés seguirà esperant una resposta de forma indefinida. Per contra, en el cas que els usuaris rebutgin la invitació, s'informa en pantalla del rebuig i l'usuari haurà de buscar nous oponents per una altra via o tornar-ho a intentar.

Rebre invitacions

Tal com s'ha vist fins al moment, mentre s'ha iniciat el mode multi-jugador l'aplicació accepta i emmagatzema les invitacions rebudes, de fet com ja s'ha comentat, aquesta és una de les raons per separar la funcionalitat del mode multi-jugador de la resta de l'aplicació i utilitzar fragments per l'activitat multi-jugador.

Així, ja sigui que l'usuari estigui jugant una partida simple (des del mode multi-jugador), com que l'usuari estigui iniciant una partida multi-jugador amb un altre usuari, al utilitzar una activitat amb fragments, per molt que es canviï el fragment visible es seguirà escoltant noves invitacions i aquestes seran en primer lloc mostrades en pantalla per tal que l'usuari pugui acceptar-la a l'instant (només si es troba al fragment principal del mode multi-jugador), o en segon lloc emmagatzemades per respondre-hi més endavant, quan l'usuari ho desitgi, ja que en cap cas no es canvia l'activitat en curs.

D'aquesta manera, des de l'opció de "veure invitacions", l'usuari pot consultar en tot moment les ofertes rebudes per part dels seus cercles d'amistats i en cas que aquests segueixin "connectats" des del mode multi-jugador (és a dir, que no hagin canviat d'activitat), aquestes podran ésser acceptades. En aquest cas l'usuari que ha enviat la proposta serà notificat i es començarà la nova partida quan aquest estigui disponible (acabi la partida en curs, etc.).

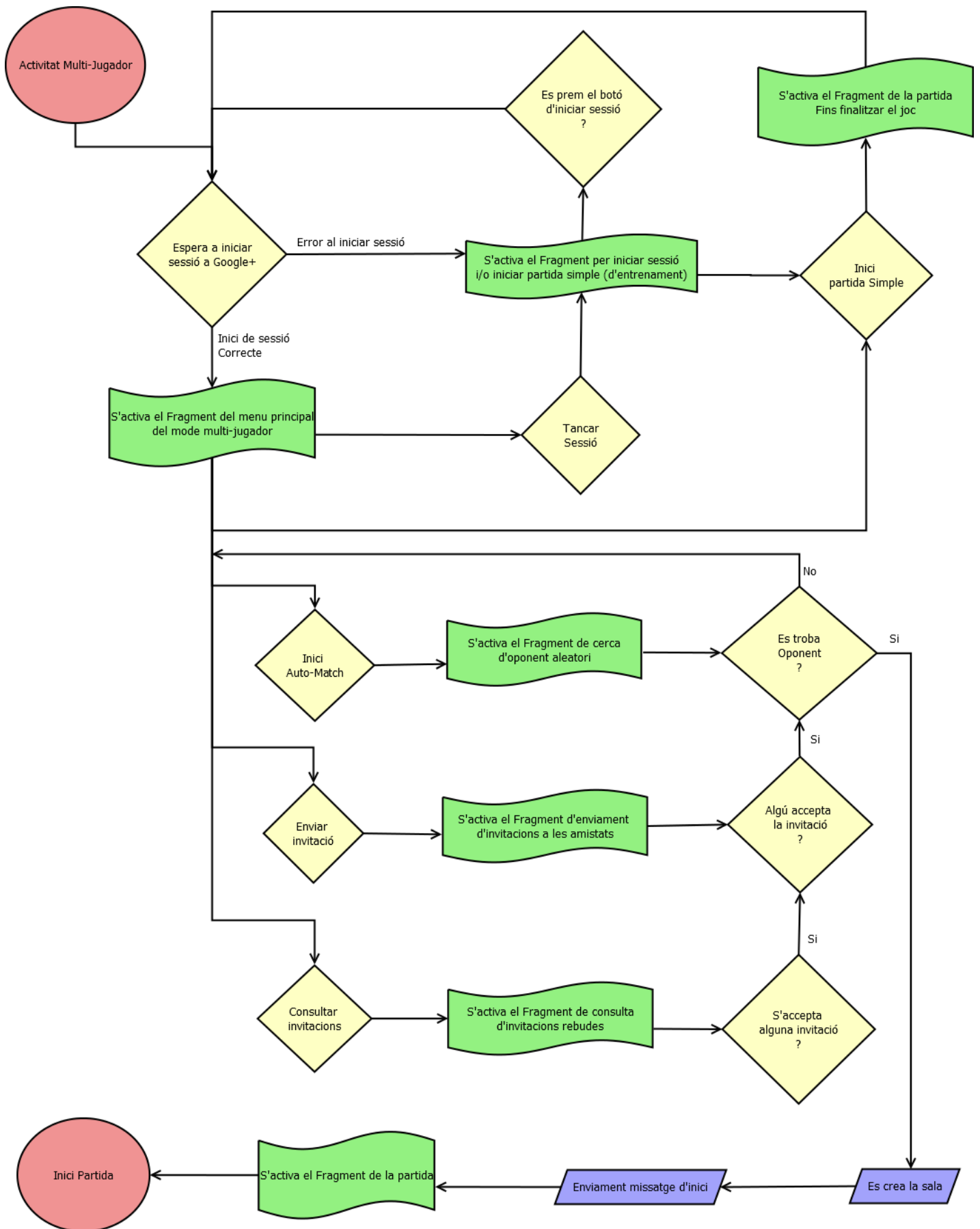


Figura 21. Diagrama de Flux – Mode multi-jugador.

3.3.1 Inici de la partida multi-jugador

Una vegada descrites totes les formes de localitzar un oponent per poder realitzar la partida, cal entendre com funciona el procés per poder començar realment aquesta partida. En primer lloc, i com ja s'ha explicat, cal que els dos oponents comparteixin una sala creada als servidors de Google prèviament per l'usuari que ha iniciat el procés d'encaixada de mans (hand-shake).

D'aquesta manera, una vegada els usuaris es troben a la sala comuna, aquesta estarà en mode "sala d'espera", fet que permet als usuaris comunicar-se directament, és a dir que, en aquest punt la comunicació ja s'ha establert.

En aquest punt és interessant que els usuaris comencin la partida al mateix instant, per això cal enviar/rebre el missatge donant el tret de sortida. Així, cada usuari envia el missatge d'inici al seu oponent, de manera que cada usuari, al rebre el missatge d'inici del seu oponent ja pot iniciar la partida. Val a dir que tot aquest procés treballa en mode "backend", ja que l'usuari realment no ha de realitzar aquestes funcions de forma manual, restant així de forma totalment transparent per als jugadors, que simplement visualitzaran que el seu oponent ha acceptat la invitació i s'iniciarà la partida amb un missatge previ d'inici (per exemple: "la partida està a punt de començar amb l'oponent X, estigues preparat").

Arribats a aquest punt, la partida està en marxa i cada jugador ha d'intentar fer-ho de forma individual millor que el seu oponent. Com que el funcionament del joc és senzill, (es tracta d'avançar el màxim possible sense topiar amb cap enemic), cada usuari jugarà la seva pròpia partida igual que si ho fes en mode de partida simple, però en tot moment podrà saber quina és la puntuació del seu oponent en cada moment, ja que queda visible en temps real a la part superior de la pantalla .

Per tant, al iniciar la partida en mode multi-jugador, l'activitat ha de crear un fil d'execució per separat que s'executi de forma cíclica durant tot el transcurs de la partida. Aquest fil d'execució (thread), s'ocupa de la comunicació entre els oponents durant la partida, i per tant ha de realitzar una consulta de la puntuació actual obtinguda per el jugador i enviar-la al seu oponent.

La raó de realitzar aquesta funció des d'un fil d'execució en paral·lel, és evitar bloquejar el fil d'execució principal que està executant la partida, ja que aquest "thread" secundari ha de, en primer lloc consultar la puntuació, en segon lloc enviar-la i finalment esperar un temps determinat "dormint" per poder tornar a executar el següent cicle, i aquest procés ha de durar mentre la partida estigui en curs i no es declari cap vencedor. Així, pot donar-se el cas que un missatge arribi en retard o que es trigui més del compte en poder enviar el missatge, però aquesta casuística no seria crítica al efectuar-se des d'un fil d'execució en paral·lel i d'aquesta manera no repercutiria en l'execució de la partida en curs.

Així, quan un dels dos jugadors topa amb algun enemic i finalitza la seva partida particular, enlloc d'enviar la puntuació de forma normal com ha fet fins al moment, aquesta s'envia amb una etiqueta que informa que es tracta de la puntuació final, i per tant l'altre usuari pot saber que el seu oponent ha finalitzat la partida. D'aquesta manera, es compara la puntuació obtinguda fins aquell moment per ambdós jugadors per tal de proclamar un vencedor i finalitzar així la partida multi-jugador.

Arribats a aquest punt si tot ha funcionat correctament ja s'ha completat la partida entre dos jugadors, però després del que s'ha vist, que succeeix si un dels dos usuaris surt de la sala o es desconnecta de forma imprevista durant la partida en curs (és a dir, sense enviar el missatge de partida finalitzada)?

Per pal·liar aquest possible imprevist o altres, GPGS disposa d'uns mètodes per controlar si s'ha perdut la connexió amb l'oponent remot sigui per la causa que sigui. Així, al iniciar una partida s'activa un escoltador que avisa a l'aplicació en cas que l'oponent surti de la sala o en cas que es perdi la connexió entre els usuaris. En aquest projecte, l'acció a realitzar en aquests supòsits serà la interrupció de la partida en curs.

Per altra banda, es pot donar el cas que l'usuari es desconnecti manualment, ja sigui de forma voluntària com perquè rep una trucada per exemple, i per tant GPGS també disposa d'uns mètodes per contrarestar aquestes casuístiques. D'aquesta manera, en el moment que es surt de la sala s'ha de comprovar si existeix una partida en curs i en cas afirmatiu finalitzar-la informant-ne a l'usuari.

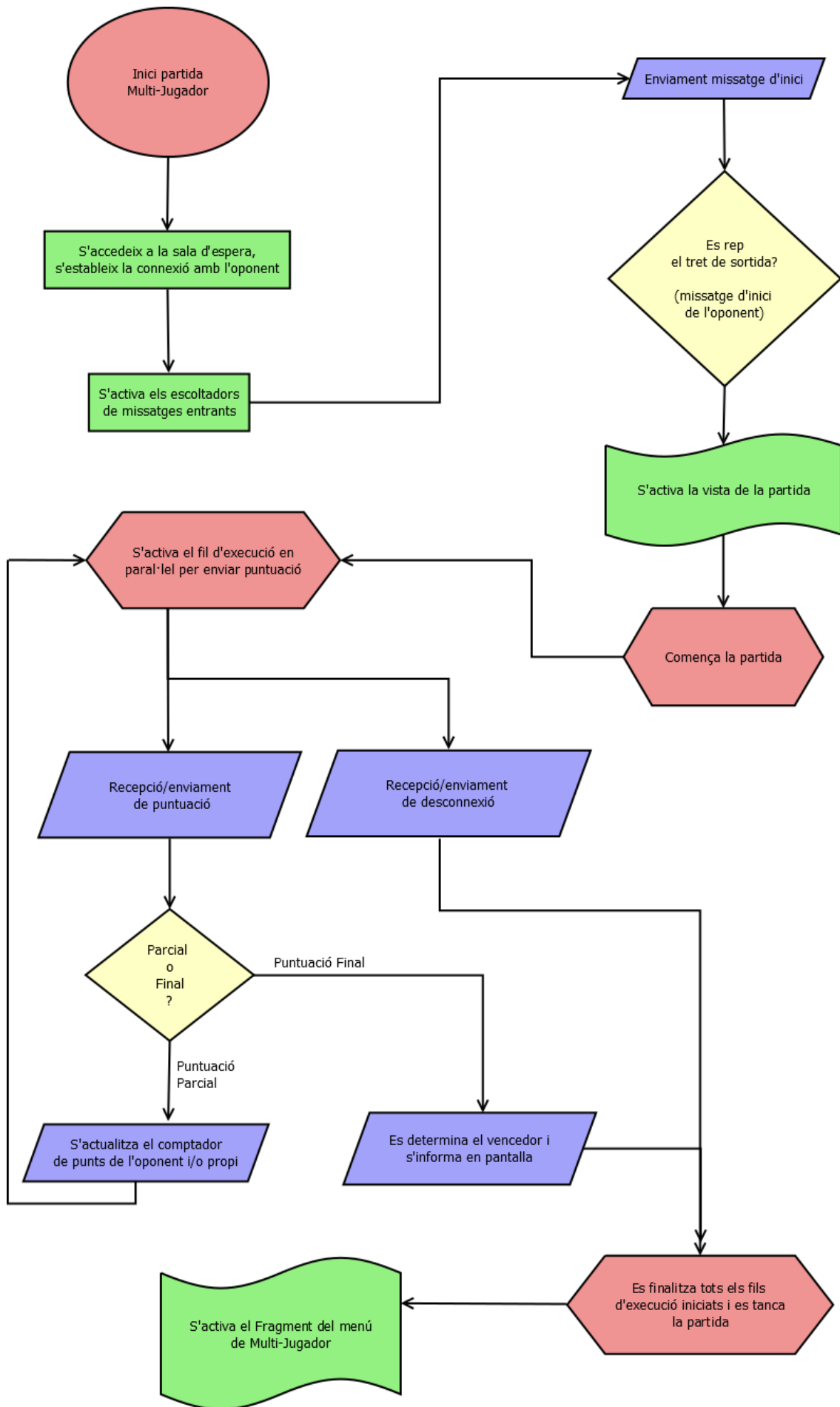


Figura 22. Diagrama de Flux – Inici partida multi-jugador.

3.4 Apartat gràfic: Renderitzat

Una vegada s'ha vist com funciona la lògica de tota la aplicació, incloent l'apartat de multi-jugador i la resta de funcionalitats del joc, es pot centrar la implementació en el que conforma el joc en si, l'execució d'una partida. Tal com s'ha vist, la partida es crida des d'una activitat, ja sigui des de l'activitat de partida simple o des de l'activitat de partida multi-jugador, però sigui quina sigui la activitat pare, sempre es crea una instància de la mateixa vista, la vista de partida. En aquest apartat doncs, es descriu el funcionament d'aquesta vista i tot el que es realitza per aconseguir renderitzar el joc en moviment, les imatges i realitzar tots els càlculs necessaris per determinar si la partida segueix en marxa o s'acaba.

Les partides es componen per una activitat pare, una vegada aquesta és creada es carrega la vista del joc, un View personalitzat, creat per gestionar tota la interfície i el comportament de la partida. Per realitzar aquest apartat es podria fer ús d'alguna llibreria o framework extern, a mode de simplificar-ne el funcionament, però en aquest cas s'ha optat per utilitzar únicament les eines disponibles per l'SDK de Android. Així, per poder començar cal entendre com funciona realment un joc en el sistema Android.

En primer lloc, tal com s'ha dit, es crea la vista de la partida i en aquesta vista (View) es desenvolupa tota l'acció de la partida. És a dir, aquesta vista gestiona les pulsacions sobre la pantalla, crea les imatges en la posició corresponent i finalment les dibuixa. Això és el que s'anomena el Loop del joc. Per entendre aquest funcionament, s'ha trobat una símil molt exemplificador en un dels recursos utilitzats per realitzar (i entendre/aprendre) aquest projecte (es referencia al final del document).

Es planteja doncs, entendre l'activitat pare (sigui quina sigui) com una taula on hi ha un paper sobre el qual es pot dibuixar, aquest paper representa la vista de la partida, per tant es pot tenir el mateix paper (vista) sobre qualsevol taula que seguirà mantenint les mateixes propietats/funcionalitats. Així, si s'agafa un llapis i es dibuixa sobre el paper, aquest llapis és la nostra font d'entrada, és a dir, representa les pulsacions sobre la pantalla i el resultat d'aquestes pulsacions es tradueix en uns traços que posteriorment es representen en mode de dibuix sobre el paper.

Així, amb aquest símil es pot entendre el procés principal d'un joc o el que s'anomena el bucle del joc. Resumint, es crea una vista amb una base pre-establerta (un personatge, un terra, uns enemics, etc.), es llegeixen les entrades (pulsacions sobre la pantalla), s'interpreten i s'actualitza l'estat/posició de les imatges en funció d'aquestes (si s'ha polsat la pantalla es fa "saltar" el personatge posicionant-lo en una posició "y" superior, es fa avançar el terra, el personatge, els enemics, etc.), i finalment, una vegada està tot preparat, es dibuixa tot d'una sobre la pantalla. Aquest procés es repeteix de forma indefinida mentre s'executa la partida i fins que no es produeix cap esdeveniment que indiqui que s'ha d'aturar (com una col·lisió).

Per altra banda, aquest cicle que es repeteix durant el joc, es podria controlar des de la pròpia vista, però com que pot resultar un procés molt feixuc, sobretot pel fet de carregar totes les imatges, calcular-ne la posició/estat corresponent i representar-les sobre la pantalla, la gestió d'aquest cicle s'ha de realitzar des de un fil d'execució en paral·lel, ja que en Android, si es bloqueja el fil d'execució principal d'una vista amb una operació feixuga es poden donar dos problemes. El primer és que la pròpia vista no sigui capaç de realitzar en temps real les seves funcions establertes (com interpretar pulsacions sobre la pantalla, etc.). El segon possible problema és que si es bloqueja durant massa temps el fil d'execució principal de l'activitat, aquesta pot deixar de respondre i enviar a l'usuari un missatge informant que la aplicació no esta responnent i s'ha de tancar.

D'aquesta manera, reprenent la explicació sobre el funcionament del joc, al instanciar la vista de la partida, aquesta inicia el fil d'execució secundari (thread), de manera que aquest s'executi de forma cíclica i indefinida. Aquest thread, es dedica doncs en primer lloc, a executar el mètode "Update" de la vista, aquest és el que calcula la posició de tots els objectes sobre el "paper", interpreta les pulsacions en pantalla, etc. En segon lloc, i una vegada calculats tots els estats i preparades totes les posicions dels objectes etc., el thread executa el mètode "Render" de la vista de la partida. El mètode "Render", agafa tots els objectes preparats anteriorment, i els dibuixa sobre la pantalla. Val a dir que per senzill que sembli, aquest punt és molt complex i costós a nivell de processament, ja que les imatges poden representar un pes i una carrega de processament molt important. S'ha de recordar i sobretot tenir molt en comte que aquest procés s'executa a cada cicle del thread secundari, i aquest es pot reproduir fins a 1000 vegades per segon en dispositius potents, però aquest fet dependrà de la freqüència del rellotge del processador de cada dispositiu.

Així, amb el que s'ha comentat fins al moment, es podria interpretar que en funció del dispositiu sobre el que s'executi el joc, aquest funcionarà més ràpid o més lent, fet que no és desitjable, a més a més s'estaria consumint una quantitat de recursos massa elevada pel que realment és necessari. Per tant, s'ha de controlar el nombre d'execucions per segon d'aquest cicle, és a dir, el nombre d'actualitzacions i renderitzats per segon, però llavors sorgeix un segon problema, però val la pena entendre correctament primer tot el procés.

Si es fixa un nombre determinat d'actualitzacions per segon (cicles del thread secundari), es poden donar diversos casos en funció de la potencia de processament de cada dispositiu, però primerament cal entendre dos conceptes, UPS (Updates/Actualitzacions per segon) i FPS (Frames/imatges per segon). Els UPS corresponen al nombre de vegades per segon que s'executa el mètode "Update" de la vista, mentre que els FPS corresponen al nombre de vegades per segon que s'executa el mètode "Render" de la vista.

El que l'usuari visualitza realment és el nombre de FPS, però per molts FPS que es renderitzin per segon, si la velocitat d'actualització és baixa el que succeirà serà que es redibuixarà varies vegades la mateixa imatge sense modificar, de manera que semblarà que el joc s'executi lentament. Per tant, s'ha de procurar establir un nombre adequat de FPS i UPS, es considera entre 20 i 30 FPS un bon nombre (al cinema es reproduïxen les pel·lícules a uns 24FPS). Així, si

es desitja mantenir uns 25 FPS, significa que s'ha d'executar el mètode "Render" cada 40ms (1000ms / 25 FPS = 40ms). Però si es te en comte que el mètode "Update" s'executa abans que el "Render", s'ha d'assegurar doncs, que el mètode "Update" s'executi en exactament 40ms, ja que si ho fa en menys temps es cridarà abans el mètode "Render" i s'obtindrà un nombre superior de FPS (més ràpid), i si tarda més de 40ms s'obtindrà un nombre inferior de FPS (més lent).

En un dels recursos utilitzats per realitzar el projecte s'explica molt clarament aquest funcionament amb uns gràfics molt entenedors i que s'adjunten a continuació. El recurs en concret es troba al següent enllaç [7]

En primer lloc, per poder entendre el que s'ha comentat, es mostra com es distribuïria el temps si es desitgés renderitzar a una velocitat de exactament 1 FPS:

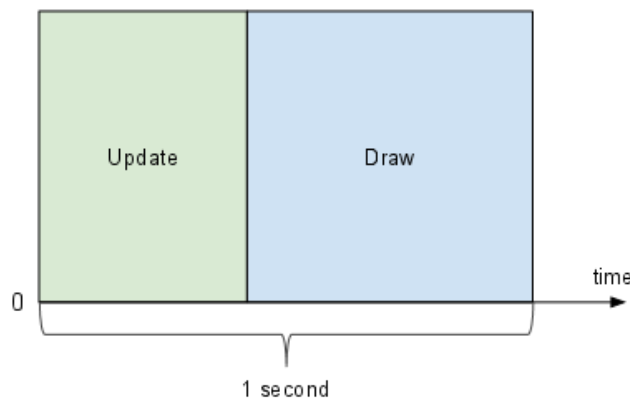


Figura 23. Renderitzat – 1FPS. [7]

Si es trasllada aquest gràfic a una escala de 10 FPS, cada cicle del thread s'executaria en 100ms:

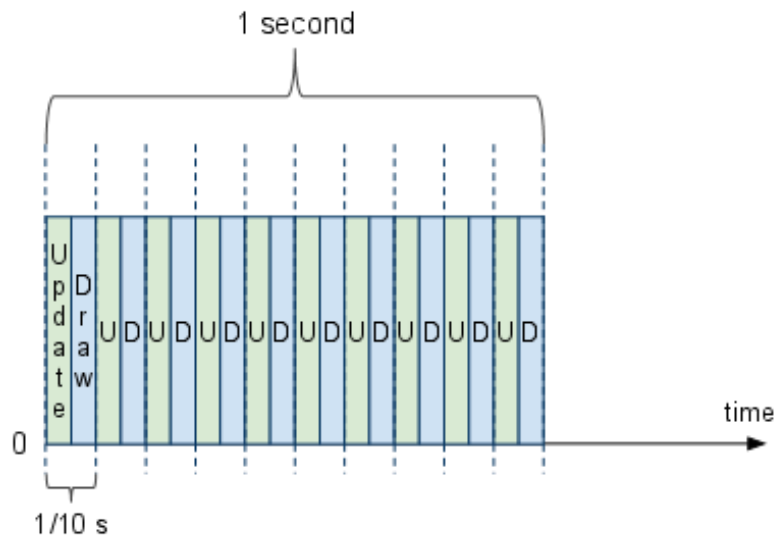


Figura 24. Renderitzat – 10 FPS. [7]

Però aquest no és el cas real, ja que en aquest cas es representaria exactament cada 100ms, però com s'ha comentat, en funció de les capacitats de processament de cada dispositiu es pot tardar més temps o menys en realitzar el cycle complet de actualització-renderitzat. Per altra banda, en el cas que només es tingues que representar 2 objectes potser aquest temps es podria escurçar i per contra si se'n tinguessin que representar 300 el temps de processament segurament augmentaria, tant per calcular-ne les actualitzacions com per renderitzar-ho. Per tant s'ha de tenir clar que aquest és un fet que a nivell de programació no es pot controlar ni preveure, no es pot determinar si hi haurà 2, 10, 50 o 500 objectes a representar a cada cycle d'actualització, per tant s'ha de trobar una alternativa que permeti fixar els FPS.

Així, si es pretén fixar els FPS a 10 (cada 100ms s'actualitza la imatge), es poden donar diverses situacions. En primer lloc pot realitzar-se el cycle complet en exactament 100ms (situació ideal en un món fantàstic, ni es plantejarà), en segon lloc es pot donar el cas que el cycle d'actualització-renderitzat acabi en menys de 100ms o finalment pot donar-se la situació que es tardi més de 100ms. Però per entendre-ho i visualitzar-ho val la pena analitzar-ho de nou amb els gràfics del recurs mencionat anteriorment, fet que permetrà entendre la solució a seguir.

Situació 2, s'acaba el cycle abans d'hora, i per tant sobra un petit període de temps en el qual el thread secundari pot aturar-se abans de continuar amb el següent cycle. Si no aturem el thread aquest temps sobrant, el joc s'executaria massa ràpid, i per tant introduint un temps de pausa s'aconsegueix en aquest cas una velocitat del joc constant:

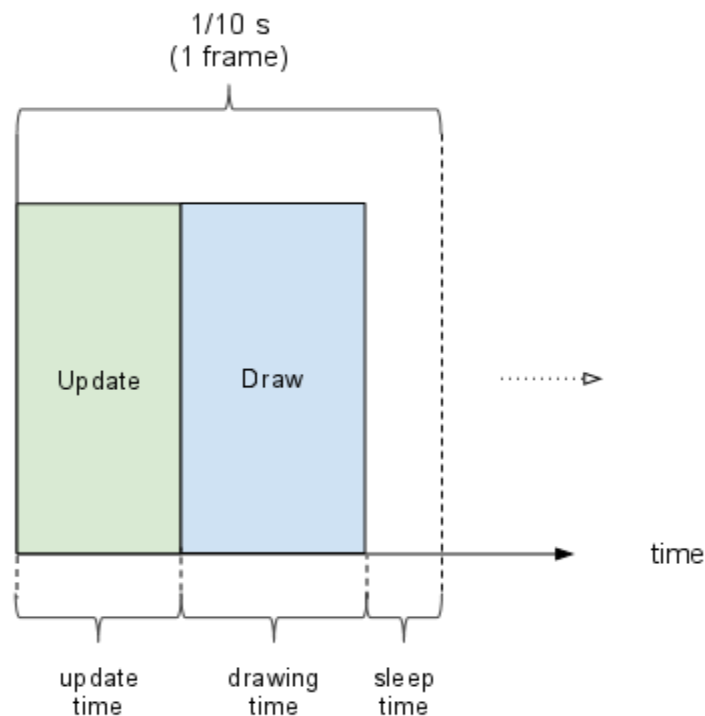


Figura 25. Renderitzat – Situació 2. [7]

Situació 3, i la més complicada de solucionar, es dona el cas que el cicle tarda més temps a executar-se del que s'ha establert:

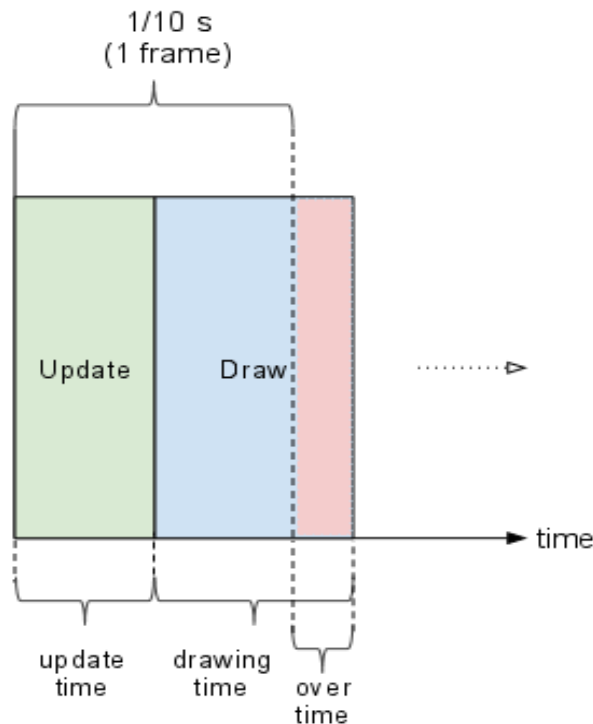


Figura 26 Renderitzat – Situació 3. [7]

Si aquesta situació es reproduïx a cada cicle, es perdrà cada vegada més temps de manera que el joc es reproduirà lentament i amb un retràs cada vegada superior a mesura que segueixi avançant. Així, en aquest cas es planteja una situació complicada, i és que el processador del dispositiu no disposa de prou capacitat per executar amb solvència el joc desenvolupat. Per tant, s'ha de trobar una solució, i de nou en el recurs mencionat es detalla la solució a seguir, bé, de fet hi ha dues opcions.

En primer lloc, s'ha d'analitzar a nivell físic el moviment, així, si es disposa un objecte en moviment a una velocitat constant i aquest tarda mig segon en recórrer mitja pantalla, es pot intuir que en el següent mig segon recorrerà la mitja pantalla restant, però aquest càlcul pot resultar feixuc i complicat d'aplicar en la física del joc.

La segona opció és realitzar l'actualització de les posicions i l'estat dels objectes (mètode Update) en intervals de temps constants, tant si s'ha pogut realitzar el renderitzat com si no, per tant, aquesta opció implica que per mantenir una velocitat de joc constant es podran descartar algunes representacions (Frames), per tant en aquest cas potser disminueixi els FPS representats a la pantalla però es manté constant la velocitat del joc. Una vegada més, tot això es pot entendre molt millor amb un gràfic extret del mateix recurs que els anteriors [7]:

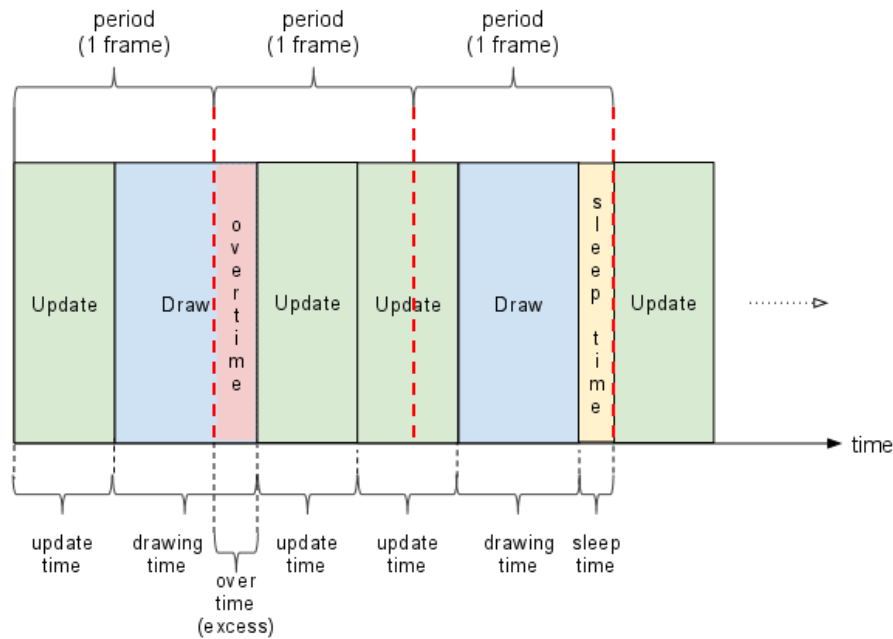


Figura 27. Renderitzat – Situació real. [7]

En aquest darrer gràfic es pot observar aquesta situació representada, on el primer cicle actualització-renderitzat s'escapa del temps establert (marca rosa "over time"), i en el següent cicle es salta el procés de renderitzat ja que en l'anterior s'ha excedit de temps i en aquest encara s'endarreriria més, per tant es tora a executar de nou el cicle de actualització tot seguit. Així, en el tercer cicle, que comença quan encara no ha acabat el procés d'actualització, ja es pot renderitzar directament tant aviat com s'acaba l'anterior "Update". D'aquesta manera en el tercer cicle acaba amb temps de sobres, el qual es posa en mode pausa al thread que ho gestiona i finalment el quart cicle pot tornar a començar amb total normalitat.

Ara bé, si la situació es torna extrema, es podria arribar a donar el cas que no es representi cap imatge nova en molta estona i aquest fet no es pot permetre, ja que es podria acabar la partida i mentre no es visualitzaria cap novetat en pantalla. Així, cal establir un màxim de Frames a "saltar-se", és a dir un màxim nombre de cicles de renderitzat que es poden evitar, si es supera el màxim es forçarà a renderitzar, fet que implicarà una davallada de la velocitat del joc, però aquesta seria una situació extrema.

Per finalitzar aquest apartat, es podria exposar de manera detallada el funcionament dels mètodes "Update" i "Render" de la vista de la partida, però es considera que aquests estan prou ben detallats i explicats en el propi codi mitjançant comentaris i resulten senzills d'entendre. Per altra banda no es pretén engreixar excessivament aquesta documentació, ja que es considera que l'objectiu d'aquesta és exposar el funcionament del joc a nivell lògic i no detallar en profunditat cada punt del codi.

Un detall que sí que cal explicar en aquesta documentació sobre el renderitzat, és com es realitza el moviment dels objectes, com ara el caminar del personatge o el "gir" de les monedes o el "volar" dels monstres enemics per posar algun exemple.

De totes maneres es un funcionament molt senzill i pot no ser l'òptim, però val la pena de comentar. Com s'ha comentat, es realitza un nombre constant d'actualitzacions per segon, a cada actualització s'activa un comptador en cada una de les classes corresponent a cada objecte (personatge, monstre, moneda i "shuriken" (estrella ninja). D'aquesta manera, cada X cicles d'actualització (en funció de cada objecte), es canvia d'estat l'objecte i en cada estat s'escull una imatge diferent. Per exemple, en el cas de les monedes la imatge corresponent al recurs que s'utilitza per representar cada moneda, està compostat per 4 imatges consecutives posicionades en una mateixa fila. D'aquesta manera, a cada canvi de cicle es representa una part diferent d'aquesta imatge. Aquest fet es pot entendre millor amb imatges:

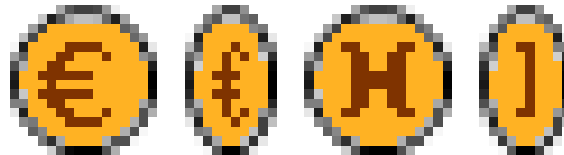


Figura 28. Gràfics del joc – Monedes.

Aquesta imatge correspon al recurs creat per al joc per representar la moneda (s'ha ampliat per poder explicar-se millor). Així en el primer estat es mostra la primera moneda, quan es canvia d'estat es representa la segona moneda, al següent canvi la tercera i finalment la quarta. D'aquesta manera es torna a començar el cicle, realitzant un efecte de gir de la moneda sobre el seu propi eix.

El mateix s'ha fet amb el personatge, la imatge del qual s'ha obtingut a partir del següent enllaç, que permet crear els personatges de forma personalitzada per utilitzar-los posteriorment en el projecte desitjat: <http://www.famitsu.com/freegame/tool/chibi/index2.html> translated



Figura 29. Gràfics del joc – Heroi.

Es pot observar que el personatge disposa de més estats que la moneda, la fila superior correspon als estats de caminar, mentre que la fila inferior correspon als estats de estar a l'aire caient o saltant. La resta d'imatges no es mostraran en aquesta documentació per no crear explicacions repetitives, ja que el concepte és el mateix.

4. Entorn de programació

4.1 Configuració Eclipse

Per la implementació del projecte s'ha utilitzat l'entorn de programació integrat (IDE) Eclipse, de totes maneres cal preparar-lo i configurar-lo per poder programar per Android, així, en aquest apartat es descriurà breument els passos a seguir per poder disposar d'un entorn adequat per poder desenvolupar per Android.

En primer lloc, cal tenir instal·lat el Java Runtime Environement (JRE) en la versió 5.0 o superior, en cas de no tenir-lo ja instal·lat a l'equip, aquest es pot descarregar en la següent pàgina web: <http://www.java.com/es/download/>

En segon lloc cal instal·lar l'Eclipse, que es pot descarregar des de la web: <http://www.eclipse.org/downloads/>

És important tenir en compte la versió del sistema operatiu que utilitza l'equip sobre el que es vol desenvolupar i escollir així la descarrega corresponent (32 o 64 bits). Pel que fa a la versió d'Eclipse es pot optar per la versió Standard (que conté totes les eines necessàries per desenvolupar i depurar amb Java) o per la versió preparada especialment per Java (IDE for Java Developers). En qualsevol cas, cal disposar d'una versió preparada per Java, ja que aquest és el llenguatge que s'utilitza en el desenvolupament per Android.

De totes maneres, l'Eclipse no requereix instal·lació pròpiament dita, simplement cal descarregar la versió corresponent i descomprimir-la en el directori desitjat. Així una vegada descomprimit ja es disposa de l'executable que permet obrir l'entorn de programació. Val a dir que al arrencar l'entorn, aquest demana un directori de treball, que serà on s'emmagatzemarà tot el codi desenvolupat.

Una vegada es disposa de l'entorn de programació, cal preparar-lo per programar per Android. Per això s'ha de disposar l'SDK d'Android a l'equip i instal·lar el complement de desenvolupament d'Android a l'Eclipse, que integra les eines necessàries a l'entorn de programació. Així, en primer lloc es descarrega l'SDK des de la web de desenvolupament d'Android: <http://developer.android.com/intl/es/sdk/index.html>

On cal elegir la opció de utilitzar un IDE existent, ja que també hi ha la possibilitat de descarregar un Eclipse ja preparat per Google en un sol paquet, de totes maneres es prefereix la modularitat d'instal·lar tots els elements per separat i així poder aprofitar el mateix entorn per altres possibles desenvolupaments no relacionats amb Android, simplement afegint els complements necessaris a l'Eclipse.

Val a dir, que l'SDK és una eina independent del IDE, ja que disposa de totes les llibreries obertes del sistema operatiu Android, de fet, l'SDK conté el codi font del sistema operatiu en si

entre moltes altres coses. Així, una de les eines disponibles amb l'SDK és l'SDK Manager, que permet descarregar els elements necessaris en cada cas. Per tant, mitjançant aquesta eina es tria les versions de Android que es vol utilitzar per desenvolupar el projecte (ja que l'SDK disposa de l'històric de totes les versions d'Android per poder descarregar i així executar-les des d'una màquina virtual per realitzar les proves durant el desenvolupament). A banda de les diferents versions del sistema operatiu, des de l'SDK Manager es pot descarregar moltes més funcionalitats i llibreries disponibles per al desenvolupament d'Android. En la següent captura de pantalla es pot observar aquesta eina:

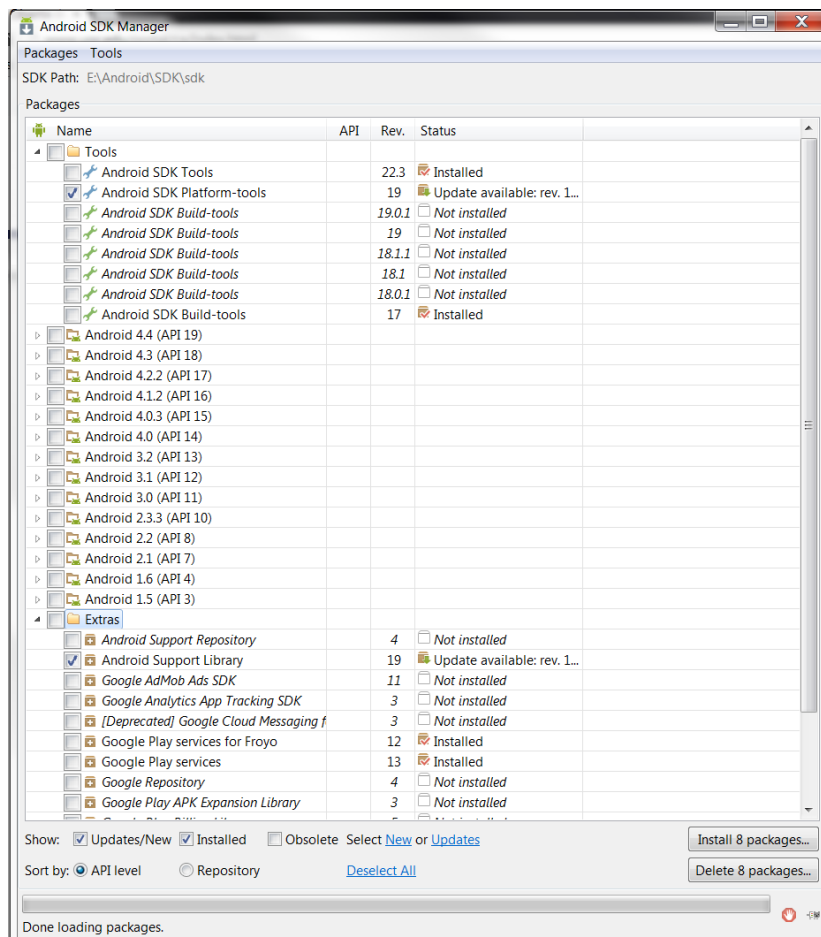


Figura 30. Configuració de l'IDE – SDK Manager.

Una vegada descarregat l'SDK en un directori conegut del disc i descarregades les funcionalitats requerides, es pot procedir amb la instal·lació dels complements de desenvolupament d'Android a l'Eclipse, que s'anomenen ADT (Android Development Tools) que en part es nodreixen de l'SDK descarregat en el pas anterior.

Així, des de l'Eclipse cal accedir al menú "Help" i triar la opció "Install New Software" i en la finestra que apareix es pot afegir un nou repositori amb el botó "Add", on aleshores s'ha de introduir la direcció que proporciona Google en les instruccions proporcionades: <https://dl-ssl.google.com/android/eclipse/>

Finalment, després d'instal·lar l'ADT i reiniciar l'Eclipse perquè la instal·lació sigui efectiva, s'ha de informar la ruta on s'ha descarregat l'SDK, d'aquesta manera l'Eclipse i el complement ADT sabran com trobar les eines disponibles per l'SDK. Així, des del menú "Window" i la opció "Preferences" s'accedeix a la finestra de configuració, on ara ja es troba una pestanya d'Android, des de la qual es pot indicar la ruta on es troba instal·lat l'SDK:

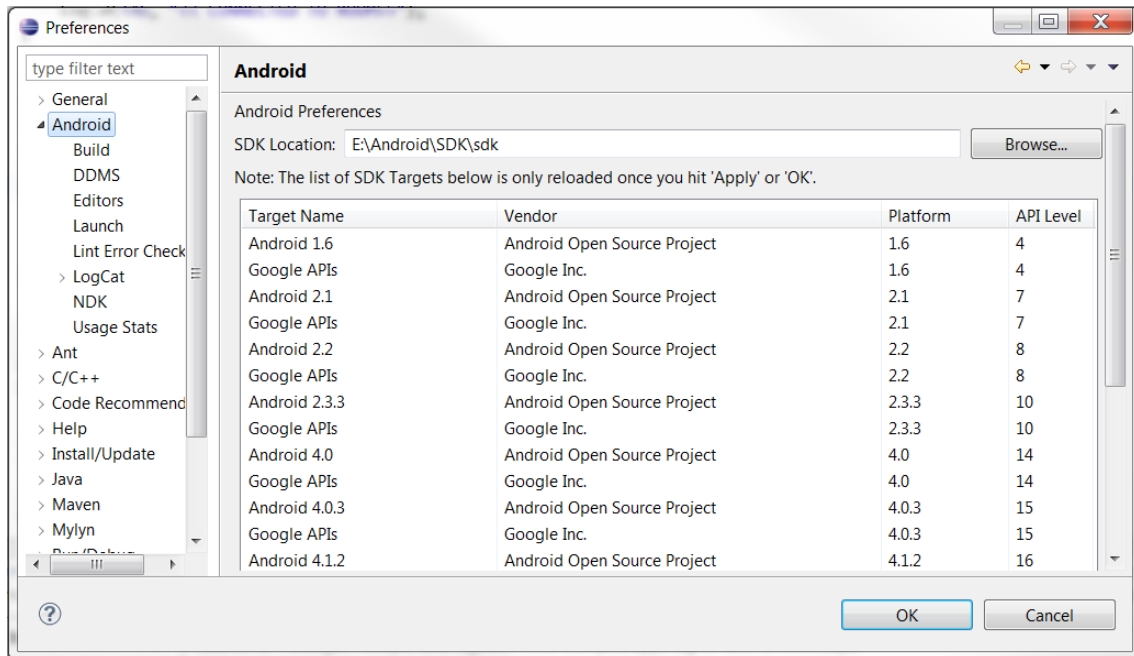


Figura 31. Configuració de l'IDE – Ubicació SDK.

D'aquesta manera, s'ha completat la configuració de l'entorn de programació per Android i arribats a aquest punt es pot crear el projecte sobre el qual desenvolupar.

4.2 Google Play Games Services

Tal com s'ha comentat durant tota la documentació, per la realització d'aquest projecte s'ha utilitzat la API de Google Play Games Services, que és un servei que dona Google als programadors per afegir funcionalitats implementades sobre "el núvol" de Google. Aquestes funcionalitats ja s'han comentat anteriorment i s'ha vist lo interessants que són, però també que no resulten trivials d'implementar. De totes maneres cal agrair fermament a Google per, en primer lloc donar aquest servei de forma (pràcticament) gratuïta als programadors (cal disposar d'un compte de desenvolupador, és a dir 25 \$), i en segon lloc pel fet de crear una documentació i exemples tan extensos per tal que tothom sigui capaç d'entendre-les i implementar-les.

Tot i que s'han comentat extensament, es recorda les funcionalitats aportades per aquesta API de Google Play Games Services:

- Rànquings de puntuacions sobre els usuaris de Google+
- Fites assolides durant el joc
- Partides en mode multi-jugador

Per poder implementar aquestes funcionalitats en el projecte s'ha seguit els passos descrits a la documentació de Google [8].

Per altra banda, s'ha utilitzat els exemples proposats per Google per tal d'implementar aquestes funcionalitats en el nostre codi. Concretament s'ha utilitzat l'exemple anomenat "BaseGameUtils" per tal d'implementar l'inici de sessió a Google+ i gestionar totes les funcionalitats relacionades amb GPGS. Aquest exemple proporciona les classes "BaseGameActivity" i "GameHelper", que són la implementació estricta dels serveis oferts per la API de GPGS i que han simplificat enormement el desenvolupament d'aquest projecte a més de permetre entendre el funcionament d'aquesta API. Val a dir que les classes esmentades disposen de llicència "Apache 2.0" i estan dissenyades amb la finalitat que els programadors les utilitzin lliurement en els seus codis.

De totes maneres, per l'apartat multi-jugador no ha estat suficient en utilitzar aquestes classes i la documentació de Google per entendre'n el funcionament. Per això s'ha basat principalment en l'exemple també proposat per Google anomenat "ButtonClicker" del repositori de Google a GitHub. Aquests exemples i altres també sota llicència "Apache 2.0" es poden trobar en el següent enllaç [9].

4.3 Fonts de codi re-utilitzat

A banda de les llibreries de Google utilitzades (GPGS) i els exemples proposats per Google, s'ha fet ús de diverses fonts de codi pre-existents per dur a terme la implementació d'aquest projecte.

En primer lloc, s'ha re-utilitzat parts de projectes propis anteriors a aquest projecte, com per exemple codi implementat durant el curs de programació d'Android Basic realitzat durant l'estiu anterior a la UPV. Les parts de codi re-utilitzades d'aquest curs corresponen a l'apartat de base de dades i el llistat de puntuacions. En aquest curs es va desenvolupar com a exemple el joc clàssic d'Asteroides, i la forma d'emmagatzemar les puntuacions s'ha implementat de la manera que es va aprendre en el curs.

Per altra banda, s'ha aprofitat també les definicions de estils creades per al joc Asteroides desenvolupat durant el curs d'Android. Finalment, també s'ha aprofitat imatges creades durant aquest, com ara les fruites utilitzades per a representar els bons d'escut que apareixen

durant la partida i que també apareixen de forma aleatòria en el llistat de puntuacions i en la botiga de bons.

La icona del joc, no ha estat creada personalment i és el logotip d'una sèrie d'animació japonesa (One Piece), de totes maneres al moment d'obtenir-la es trobava sota llicència Creative Commons 3.0 (lliure utilització per a finalitats no comercials), però actualment la llicència ha expirat, de totes maneres la imatge s'ha extret del següent enllaç: http://es.todo-anime.wikia.com/wiki/Archivo:Luffy_flag_by_zerocustom1989.png

Pel que fa a la estructura del bucle del joc i el thread secundari, s'ha après a fer mitjançant el recurs ja comentat anteriorment en l'apartat de renderitzat [7].

De totes maneres, val a dir que aquest ha estat el complement que ha permès acabar d'entendre el que s'ha après a partir de les vídeo-guides realitzades per "Linus Granath", que han estat la brúixola que ha guiat gran part del desenvolupament i l'aprenentatge, aquestes es poden trobar al següent enllaç al seu canal de "YouTube" [10].

5. Conclusions

Durant la elaboració del projecte s'ha realitzat un joc del tipus Shoot'em Up amb capacitat multi-jugador per a dispositius Android, complint així amb els objectius fixats i afegint funcionalitats socials que no estaven previstes en el plantejament inicial.

Així, el desenvolupament d'aquest projecte ha estat molt gratificant alhora que complicat, fet que ha permès un aprenentatge extens sobre la programació per Android i el desenvolupament de jocs mitjançant les eines proporcionades per l'SDK i altres d'externes com les llibreries de GPGS.

Tot i que ja es disposava prèviament d'uns coneixements bàsics pel que fa al desenvolupament per Android, gràcies a les capacitats adquirides durant la carrera, la elaboració aquest projecte ha permès assolir-ne de nous amb agilitat i capacitat de reacció davant els contratemps.

Durant la implementació del projecte, tal i com s'ha exposat en aquesta documentació, han sorgit alguns imprevistos que han implicat un replantejament i un canvi de rumb sobre el disseny i la planificació plantejats inicialment, però s'ha pogut reaccionar a temps, adoptant una estratègia alternativa i escurçant la planificació temporal dels apartats següents.

Així, finalment s'ha assolit els objectius en el temps establert, tot i modificar el disseny inicial, on es plantejava utilitzar el Framework gràfic LibGDX i en el seu lloc s'ha utilitzat la classe Canvas de l'SDK d'Android. Val a dir, que la implementació mitjançant LibGDX hauria pogut ser molt interessant alhora que eficient, però com ja s'ha exposat en aquesta documentació, el temps disponible no ho ha permès.

Tot i així, l'aprenentatge assolit amb la utilització del Canvas per al renderitzat dels gràfics i la lògica d'aquest mitjançant el bucle del joc, creat a partir d'un thread secundari per mantenir estable la velocitat d'execució, ha resultat molt satisfactori i enriquidor a nivell personal alhora que resultant a nivell funcional. Per tant, també es valora molt positivament la capacitat de reacció adquirida i demostrada en trobar i aplicar una alternativa viable mantenint i respectant relativament els terminis estipulats en la planificació.

Finalment doncs, s'ha aconseguit desenvolupar un joc funcional, entretingut i amb certa càrrega addictiva, mentre que, a banda del resultat final obtingut, s'ha adquirit molts coneixements nous, tant pel que fa al funcionament i desenvolupament del motor gràfic d'un joc, com en l'ús de les llibreries de Google Play Games Services (utilitzades en les funcionalitats socials i el mode multi-jugador).

5.1 Línies de treball futur

Amb tot el que s'ha exposat en aquesta documentació el projecte encara no es pot donar per acabat al 100%, ja que es podria millorar o ampliar notablement. Així, tot i que el joc funciona, és estable i compleix els objectius fixats en un principi, encara pot donar més de sí. Per altra banda, no només a nivell de funcionalitats i implementació estrictament parlant, sinó que tampoc està acabat l'apartat visual, ja que el joc no està traduït en els idiomes més comuns, fet que caldria realitzar per publicar-lo així a la botiga d'aplicacions de manera que el joc pugui arribar a un públic major.

Per altra banda, el mode multi-jugador es podria millorar i ampliar, permetent nous tipus de partides en línia, com per exemple una partida col·laborativa, on enlloc de competir per la millor puntuació, els dos jugadors formarien un tàndem d'herois per aconseguir puntuacions majors o per guanyar partides molt difícils. Pel que fa als duels del mode multi-jugador actual, es podria millorar la funcionalitat, permetent al jugador que mor primer, seguir veient la partida del seu oponent, permetent així al seu oponent seguir amb la partida en curs.

Per altra banda, valdria la pena reprendre les investigacions sobre LibGDX i acabar desenvolupant el joc mitjançant aquest Framework per tal de permetre millores en el renderitzat i el rendiment del joc, augmentant l'estabilitat sobretot en dispositius de resolució molt gran. Com s'ha vist en el transcurs d'aquest projecte, aquest fet implicaria un cost temporal elevat, però podria resultar en una visibilitat i jugabilitat més aconseguida, amb moltes més opcions i una física del joc més complerta, per exemple es podria implementar una millora per atraure monedes, o crear patrons de moviment dels enemics de manera molt més senzilla.

Així, es podrien implementar noves funcionalitats que enriqueixin la jugabilitat i permetin un entreteniment més durador, és a dir, aconseguir que el joc enganxi més a l'usuari.

Una possibilitat interessant seria la d'afegir nivells d'experiència al personatge, de manera que la realització de cada partida estigui recompensada amb un increment d'experiència, que a la llarga permeti augmentar el nivell del personatge. A cada augment de nivell es podria millorar algun aspecte del personatge, com la velocitat d'aquest, el nombre de salts disponibles, la resistència enfront els atacs enemics o les puntuacions per metre avançat atorgades per posar-ne algun exemple. Per altra banda, en funció del nivell del personatge es podria jugar partides adequades al nivell d'aquest, implementant nous enemics més resistents o amb capacitat de disparar per exemple.

Una altra possibilitat, a banda dels nivells del personatge podria ser implementar diferents nivells de joc, permetent a l'usuari escollir (o tenir que guanyar-se el dret) participar en una pantalla senzilla amb un nivell de dificultat baix o en una pantalla difícil, on podrien aparèixer enemics amb més freqüència, amb més resistència o amb l'habilitat de disparar per exemple.

La botiga de bons podria ser objecte de una remodelació completa, millorant-ne no només la interfície gràfica, sinó permetent l'adquisició de noves funcionalitats. Un exemple podria ser la funcionalitat d'adquirir nous moviments per al personatge, com poder avançar o retrocedir per

la pantalla durant una partida mitjançant una palanca de moviment (Joystick). En la botiga també es podria permetre la compra d'armes, amb millores de potència o velocitat d'aquestes, per exemple permetent disparar en mode metrallera (mantenint polsada la pantalla es dispararia una ràfega de bales), o una arma que permeti disparar en l'angle desitjat (no només en línia recta).

Totes aquestes possibilitats podrien implementar-se juntament, plantejant un model d'experiència i nivells que bloquegin determinades adquisicions o inclús permetent la compra de millores mitjançant diners reals a través del que s'anomena pagament "in app" (des de l'aplicació).

Finalment, un tret important a millorar són les imatges del joc, que resulten senzilles i amb poc encant. Es podria buscar ajuda d'un dibuixant (o almenys d'algú amb millors dots artístiques) per crear uns enemics més variats i amb més encant per exemple, o un heroi realment impressionant que sigui agradable a la vista (cal recordar que l'heroi utilitzat s'ha creat mitjançant una web de creació de ninots en molt poca resolució per utilitzar en la creació de jocs de caire "retro"). En aquest projecte no era l'objectiu exposar els dots artístics de l'autor i per tant no s'ha treballat excessivament l'apartat del dibuix, però és important millorar-ho de cara al futur ja que els gràfics d'un joc són la primera impressió que s'enduu l'usuari.

Es poden resumir les línies de treball futur en ens següents punts:

- Traducció
- Millores Multi-Jugador
 - o Mode de partida col·laborativa
 - o Seguir visualitzant la partida del oponent
- Renderitzat i motor gràfic del joc amb LibGDX
- Física més extensa
- Noves funcionalitats
 - o Nivells d'experiència
 - o Nivells de dificultat (Pantalles i/o Enemics)
- Ampliar la botiga de bons
 - o Millores de moviment (velocitat, nous moviments, etc)
 - o Resistència (vides)
 - o Bons de puntuació per metre
 - o Noves armes
- Pagaments reals "in app" (per adquirir monedes virtuals)
- Aspecte gràfic – Dibuixos
- Publicació al Play Store

6. Glossari

A

Android: Sistema operatiu per a dispositius mòbils creat per Google

Activity: Cada pantalla d'una aplicació Android

ADT: Acrònim de Android Development Tools

Android Studio: Entorn de desenvolupament integrat creat per Google especialment adaptat per Android.

API: Acrònim de Application Programming Interface (interfície de programació d'aplicacions). És un conjunt de funcions i procediments que permeten una capa d'abstracció a nivell de programació.

App: Abreviatura de aplicació (del anglès Application).

B

BlackBerry: Dispositius mòbils de l'empresa RIM (Research In Motion, actualment l'empresa és BlackBerry), es caracteritzen per utilitzar el sistema operatiu propi RIM OS

C

Canvas: Classe d'Android que representa una superfície on dibuixar. Permet el renderitzat de figures en 2 dimensions.

F

Framework: Conjunt d'eines que permeten programar utilitzant un determinat llenguatge permetent una capa d'abstracció pel programador per tal de simplificar tasques comunes en gran part de desenvolupaments.

G

GPGS: Acrònim de Google Play Games Services, una llibreria proporcionada per Google per tal de utilitzar els serveis que aquest posa a disposició dels programadors per vincular els jocs amb els serveis de la xarxa social Google + i implementar funcionalitat

multi-jugador, l'assoliment de fites en els jocs, puntuacions compartides amb la xarxa social, entre altres.

I

IDE: Acrònim de Entorn de Desenvolupament Integrat (de l'anglès Integrated Development Environment).

iOS: Sistema operatiu per a dispositius mòbils creat per Apple.

J

JRE: Acrònim de Java Runtime Environment.

L

Layout: Fitxer en format XML que defineix l'aspecte visual d'una activitat d'Android.

M

Motor gràfic: Lògica de renderitzat i moviment dels gràfics d'un joc.

P

Play Store: Botiga d'aplicacions per a dispositius que utilitzen el sistema operatiu Android.

R

Renderitzat: Procés de representació dels gràfics en pantalla.

S

SDK Android: Acrònim de Software Development Kit (Kit de desenvolupament de software), que dona obertament Google per al desenvolupament d'aplicacions per Android.

Shoot'em Up: Tipus de joc en que un personatge avança de forma indefinida per una pantalla, on apareixen enemics que s'ha de disparar i/o esquivar.

Smartphone: Dispositiu mòbil intel·ligent.

SO: Acrònim de Sistema Operatiu.

Store: Traducció de botiga en anglès, utilitzat per referir-se a les botigues d'aplicacions dels sistemes operatius mòbils.

T

Thread: Fil d'execució o sub-procés secundari. Permet realitzar diverses tasques alhora utilitzant les capacitats multi-tasca dels sistemes operatius.

W

Windows Phone: Sistema operatiu per a dispositius mòbils creat per Microsoft.

7. Bibliografia

1. Gironés, J.T., *El Gran libro de Android*, Marcombo, Editor, 3a edició - 2013
2. Google. *Documentació oficial d'Android*. [cited 2013; Recurs web]. Available from: <http://developer.android.com>.
3. *Edu4Java*. [cited 2013; Video Guies]. Available from: <http://www.edu4java.com>.
4. Steigert, G. *LibGDX – Steigert - Android Development*. [cited 2013; Web blog]. Available from: <http://steigert.blogspot.com>.
5. Google. *Documentació oficial de Google*. [cited 2013; recurs web]. Available from: <http://developers.google.com>.
6. Google. *Gràfics de versions Android*. [cited 2013; Informació i gràfics realitzats per Google, sota llicència Creative Commons Attribution 2.5]. Available from: <http://developer.android.com/intl/es/about/dashboards/index.html>.
7. Jano, T. *The Game Loop*. [cited 2013; Web blog]. Available from: <http://obviam.net/index.php/the-android-game-loop/>.
8. Google. *Google Play Games Services*. [cited 2013; Recurs Web]. Available from: <https://developers.google.com/games/services/>.
9. Google. *Exemples GPGS*. [cited 2013; Repositori GitHub de Google]. Available from: <https://github.com/playgameservices/android-samples>.
10. Granath, L. *Android Endless Running Game*. [cited 2013; Video Guies - Canal YouTube]. Available from: <http://www.youtube.com/watch?v=gx7bxb2TUg&feature=c4-overview-vl&list=PL5xhT8WWHZ9gTAiJgJrHEjOQefcdJi5az>.