

Memòria projecte: Aeon Space Fighter (videojoc per dispositius Android)

Alejandro Santiago Varela

Enginyeria Informàtica

Prof. Roman Roset Mayals

Barcelona, 06-06-2012

Dedicatòria i agraïments

M'agradaria donar les gràcies a les següents persones.

Prof. Roman Roset Mayals (Universitat Oberta de Catalunya). Per l'ajuda a l'hora de guiar el meu projecte i el suport en la primera fase del projecte des de l'idea inicial, al projecte final. Gràcies.

A la meva família: Per tots els anys de suport des de els inicis fins arribar a ser Enginyer Superior en Informàtica. Gràcies per creure en mi.

Estela: Gràcies, sense tu, això no hauria estat possible.

To all of you that believe in me during these years. Thank you.
Per tots aquest que heu cregut en mi durant aquest anys. Gràcies.

Descripció

Àrea Programació d'aplicacions mòbils.

Android és un sistema operatiu mòbil basat en Linux que ha estat desenvolupat per ser usat en dispositius mòbils com telèfons intel·ligents, tabletas i Google TV entre d'altres. Està sent desenvolupat per una organització (Open Handset Alliance) liderada per Google.

Aquest sistema operatiu ha esdevingut una de les plataformes més famoses per desenvolupament en dispositius mòbils juntament amb IOS de Apple (I-Phone, I-Pad, etc). Actualment compta amb una gran comunitat de desenvolupadors creant aplicacions per aquest plataforma, i estenent així les funcionalitats de la mateixa. Donat que el llenguatge de programació és Java, fa que aquesta plataforma hagi esdevingut molt popular últimament, per la facilitat en que molts desenvolupadors poden aplicar els seus coneixements en aquest llenguatge a la nova plataforma.

El projecte consistirà en un videojoc 2D per la plataforma de Android (Google). Es desenvoluparà utilitzant la implementació nativa de OpenGL de la que disposa aquesta plataforma. Aquest joc despondrà de les funcionalitats més típiques, com pot ser portar els punts del jugador, un inici i un final, una lògica, musica, mètode d'entrada de les instruccions per part del jugador, etc.

En resum el projecte sorgeix com a iniciació per introduir-me en el món de la programació de dispositius mòbils, en una de les dues plataformes més grans, la de Android de Google. També hi ha l'industria del oci digital, que factura cada any més que la del cinema i la música juntes, i és una de les que millor ha resistit la crisi global que estem patim els últims anys. Per tant, ajuntant aquestes dues idees, l'objectiu d'aquest PFC serà la de desenvolupar un videojoc per Android en 2D per tal d'aprofundir els meus coneixements en aquestes dues àrees. Per últim també m'agradaria remarcar, la meva passió per la programació de videojocs com a afició, i el especial interès que tenia en el desenvolupament d'aquest PFC, per tal d'expandir els meus coneixements en aquest camp.

El projecte s'ha publicat com a programari lliure, les futures actualització i millores es poden descarregar a l'adreça:

- git de github amb el codi font:
<https://aeonphyxius@github.com/aeonphyxius/AeonSpaceFighter.git>
- pàgina del projecte a github:
<https://github.com/aeonphyxius/AeonSpaceFighter>

Paraules clau:

Android, mòbil, programació per telèfons intel·ligents, programació de videojocs, Java, Open GL ES, ADT.

Continguts

Dedicatòria i agraïments	2
Descripció	3
Continguts	4
Introducció	6
Justificació i objectius del PFC	6
Justificació	6
Funcionalitat de les diferents parts del PFC	7
Mètode de desenvolupament.	7
Enfocament i mètode seguit	8
Productes obtinguts.....	8
Descripció dels següents capítols de la memòria.	9
Planificació del projecte.	10
Cicle de vida del projecte	10
Planificació de les diferents tasques	11
Descripció de les tasques.	12
Desviaments del pla	14
El context.	15
Introducció i història dels videojocs.	15
Introducció als videojocs.....	15
Història	15
El mercat del videojoc	17
La cultura dels videojocs	17
Android	18
Història d'Android	19
Google Play.....	21
Anàlisi del sistema	23
Diagrama de casos d'ús	23
Descripció dels casos d'ús	25
JU01 Iniciar Videojoc	25
JU02 Sortir del videojoc	25
JU03 Accedir a les opcions del videojoc	26
JU04 Modificar so (SI/NO)	26
JU05 Modificar vibració (SI/NO)	27
JU06 Modificar nivell de dificultat	27
JU07 Jugar.....	27
PN01.....	28
PN02.....	28
PN03.....	28
IA01	29
IA02	29
IA04	30
IA05	30
Descripció de les tecnologies utilitzades	31
Java	31
ADT (Android Development tools)	31

OpenGL ES	32
Disseny	33
Arquitectura global	33
Decisions tecnològiques	34
Diagrama de classes.....	35
Dades	35
Objectes pintats per pantalla	36
Manegadors de conjunt d'elements (Manager).....	37
Components de la interfície gràfica d'usuari del video joc.....	38
Pantalles de transició dintre del joc.....	39
El so	40
Nucli de l'aplicació.....	41
Pantalles o "Activity" del videojoc	42
El desenvolupament	43
Instal·lació i configuració del entorn de programació	43
Requisits previs	43
Instal·lar Java	43
Instal·lar Eclipse	43
Instal·lar ADT (Android Development Tools) i l'extensió per Eclipse	44
Implementació	45
Les diferents pantalles del videojoc.....	45
El jugador	50
Els enemics.....	50
La interfície d'usuari	51
El so	55
Publicant el projecte en el "Google Play"	56
Registrar-se com a desenvolupador oficial	57
Preparant el binari per la seva distribució	57
Signant l'APK resultant del projecte	58
Publicant Aeon Space Fighter en "Google Play"	58
Seguiment de l'aplicació.....	60
Resum	61
Conclusions.....	62
Bibliografia	64
Els llibres	64
Annex	65
Com fer servir el videojoc.....	65
Samsung Galaxy SII.....	65
Emulador	65

Introducció

Aquest apartat servirà d'introducció de la memòria del Projecte Final de Carrera. Provaré de justificar la meua tria del mateix, el context en el que l'he desenvolupat, la planificació que faré servir, i la descripció general d'aquest document.

Justificació i objectius del PFC

A continuació justificaré a la tria del Projecte Final de Carrera (àrea, tipologia de projecte, etc) i els objectius del mateix.

Justificació

Aquest Treball Final de Carrera es realitzarà en el context del àrea de programació de mòbils, amb el objectiu final del mateix serà la creació d'un videojoc per una de les plataformes tecnològiques, la de Android de Google, que passa per ser una de les dues més grans que es troben disponibles en el mercat actualment.

Per donar-li un major sentit i coherència al projecte, procediré en el mateix com si d'un projecte professional es tractés. Primer de tot serà definir els requisits del mateix, per tal de donar-li forma a la idea inicial. Un cop definits els requisits inicials, el pla per portar-lo a terme, començarà amb la fase d'implementació amb un seguiment del projecte (en la forma de les diferents PACs), fins a dur a terme el producte final del mateix, el videojoc i la memòria.

Les bases tecnològiques del projecte en les que es basarà la solució final del mateix, han estat triades entre les dues opcions més grans que hi ha al mercat ara mateix, Android i IOS. Per una major afinitat amb el llenguatge de programació (Java) de la plataforma d'Android, i un desconeixement de Objective C em va fer decantar-me per la plataforma d'Android. També la possibilitat de provar el resultat en el meu telèfon Android va fer més atractiva aquesta plataforma per motius personals.

En resum la meua tria d'àrea del PFC va ser principalment per introduir-me en el desenvolupament per aplicacions mòbils i les ganes d'aprendre aquesta tecnologia, i les opcions que em pot obrir en el futur. A continuació per triar la temàtica del mateix em vaig guiar per una afició personal, que des de fa molts anys m'apassiona el món del desenvolupament de videojocs. Un cop triat el àrea i la temàtica del mateix, només en quedava convèncer al consultor (gracies Roman), per tal de poder desenvolupar la meua idea inicial del mateix.

Funcionalitat de les diferents parts del PFC

L'objectiu del projecte és dissenyar i desenvolupar un petit videojoc per una de les plataformes més famoses del mercat (Android). Així doncs el resultat del PFC serà un servidor videojoc desenvolupat en Java, l'API d'Android i OpenGL ES de manera nativa.

El videojoc haurà de permetre aquestes operacions bàsiques:

- Permetre el joc en mode 1 jugador.
- Rebre esdeveniments d'entrada i sortida del usuari.
- Reproduir musica i so.
- Fer vibrar el dispositiu mòbil, depenent dels esdeveniments al videojoc
- Jugar a diferents nivells per assolir el final del videojoc (i totes les interaccions humà-maquinari per dur a terme quest punt).

Mètode de desenvolupament.

La tècnica que s'utilitzarà serà la de l'orientada a objectes (OO), fent el desenvolupament en un llenguatge orientat a objectes. En aquest cas es farà servir Java per tot el desenvolupament, i OpenGL ES per manejar els gràfics i Android com a plataforma hardware. Els objectius seran:

- Conèixer el funcionament del llenguatge Java per Android (i les llibreries que es poden usar en aquest tipus de dispositius).
- Aprofundir en la metodologia de desenvolupament de videojocs en plataformes mòbils.
- Aprofundir en el coneixement de la tecnologia OpenGL ES (i les característiques especials en dispositius mòbils).

Es seguirà el cicle de vida del desenvolupament de programari inspirat en el "Rational Unified Process":

- Recollida de requisits: especificació de la funcionalitat que se li vol donar a l'aplicació, tipus de dades que haurà de manipular, tipus usuaris que utilitzaran l'aplicació.
- Anàlisi i disseny: creació de les classes i diagrames que representaran l'esquelet de l'aplicació, amb el llenguatge UML i amb l'eina Microsoft Visio.
- Realització: implementació de l'aplicació.

- Proves i publicació: Un cop finalitzat el procés de desenvolupament, es portarà a terme un conjunt de proves per validar el videojoc, i és publicarà en Google Play

Enfocament i mètode seguit

Un cop fet l'estudi previ, el següent pas ha estat la definició dels requisits de la nostre aplicació (definició de l'arquitectura). Així doncs, s'ha definit tota la funcionalitat del videojoc. Aquesta definició s'ha dut a terme pensant en la versió 1.0 del producte. Son uns mínims per a la primera versió i per tal de completar el TFC,

El desenvolupament de programari s'ha fet seguint un mètode de desenvolupament iteratiu e incremental, per tal de poder disposar d'una aplicació funcional tan aviat com fos possible i poder corregir els possibles errors de disseny introduïts. Això s'ha dut a terme desenvolupant el esquelet del videojoc, i fent servir per les proves un dispositiu Android del que ja disposava i el emulador.

Productes obtinguts.

Els productes obtinguts principalment han estat dos, el videojoc i tota la documentació de la memòria. A més a més hi ha d'altres productes secundaris. A continuació veurem una descripció tots els productes obtinguts.

- Videojoc: És el producte final del projecte. Construir una aplicació compatible amb mòbils Android (instal·lable apk). Hi haurà per una banda el codi font del videojoc en l'entrega final, i al rebost públic de GitHub. Per l'altre banda s'entregarà també el instal·lable del videojoc en format apk.
- La Memòria / documentació: Consisteix en per una banda aquest document com a memòria del projecte. Per altre banda als documents de les diferents PACs de las que ha consistit el projecte. A més a més hi hauran d'altres documents, com el origen del pla del projecte (Microsoft Project), diferents manuals que s'han fet servir en el desenvolupament d'aquest projecte, etc. La distribució de la documentació serà la següent:
 - **Links:** Aquí s'emmagatzemen els links que vaig visitant amb informació util, per tal de tenir un historia.
 - **Manuals:** Els diferents manuals que han sigut d'utilitat per el desenvolupament del projecte.
 - **PACs:** Les 4 PACs de les que consta el PFC. Es podria dir que son com informes del estat.
 - **Presentacio:** Presentació final del projecte. Serà una explicació breu del projecte, a mode de petita introducció.
 - **Schedule:** Pla del projecte en format de Microsoft Project.
 - **Videos:** Vídeos que s'han produït del videojoc funcionant, durant algunes de les practiques, i el que acompanyarà l'entrega final.

- El joc de proves: Resum de les proves que s'han dut a terme per tal de fer els tests necessaris per comprovar el correcte funcionament del videojoc.
- Vídeo de l'aplicació funcionant: Diferents vídeos per tal de mostrar el videojoc en funcionament sense necessitat d'instal·lació o compilació.
- Presentació del projecte: Una petita presentació del resultat del projecte, explicatiu de tot el procés.

Descripció dels següents capítols de la memòria.

- **Planificació.** En aquest capítol es tractarà tot lo relacionat amb la planificació del projecte, el pla, la definició de les tasques a dur a terme, etc.
- **El context.** En aquest capítol descriure tot lo relacionat amb el context en el qual es situa el projecte, es a dir, que son els videojocs, altres jocs disponibles, tendències de mercat, etc.
- **Anàlisi del sistema.** Els requisits previs del sistema, els casos d'ús i la descripció de les tecnologies que farem servir en aquest projecte formaran part d'aquest capítol. Es parlarà dels requisits del sistema a priori, i algunes de les funcionalitats del mateix.
- **Descripció de les tecnologies.** En aquest capítol veurem una descripció de les tecnologies que farem servir per el projecte.
- **Disseny.** En aquest capítol, es definirà tota l'arquitectura del sistema, decisions tecnològiques sobre el desenvolupament, el diagrama de classes del projecte.
- **El desenvolupament.** Tot lo relacionat amb el projecte. Definició, estructura, joc de proves, disseny, problemes sorgits en el desenvolupament. També es definirà la instal·lació del entorn de programació.
- **Conclusions.** Conclusions obtingudes després de la finalització del projecte. Resum de tot lo vist relacionat amb la programació per dispositius mòbils / videojocs.
- **Glossari.**
- **Bibliografia.**
- **Apèndix.** Tota la informació addicional, com pot esser la configuració del entorn de programació, com fer servir els productes de la memòria.

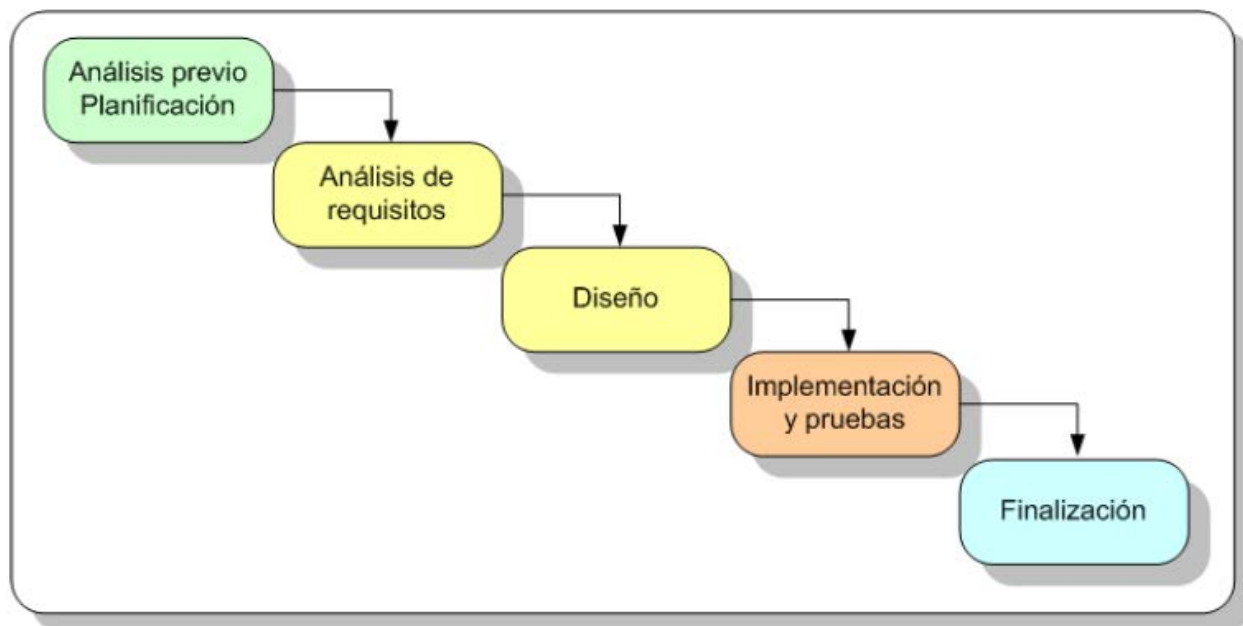
Planificació del projecte.

En aquest punt podrem observar la planificació inicial, de la qual veurem una descripció exhaustiva, i com ha variat durant el temps, ja que per diferents motius he hagut de canviar una mica el planificació inicial.

Cicle de vida del projecte

El projecte de desenvolupament del videojoc, com a qualsevol altre projecte de desenvolupament de programari, serà desenvolupat prenent com a base el cicle de vida clàssic de creació de programari o també conegut com a cicle en cascada. Ha estat adaptat a les necessitats de temps i requeriments del Projecte Final de Carrera, i al punt que sóc només una persona en el desenvolupament del mateix.

Com es pot observar en el gràfic següent, el projecte s'inicia en un anàlisis previ i la planificació del mateix (punt en el que estem ara mateix), a continuació analitzarem els requisits del videojoc, farem el disseny, continuarem amb la fase d'implementació del programari i les proves del mateix, per terminar amb la finalització del producte.









Planificació 1

Planificació de les diferents tasques

La planificació del projecte, com es pot veure en gràfic més següent, quedarà dividida en 4 grans fases (Recollida de requisits / Documentació, Anàlisi i disseny de l'aplicació, Implementació del videojoc, Memòria i entrega).

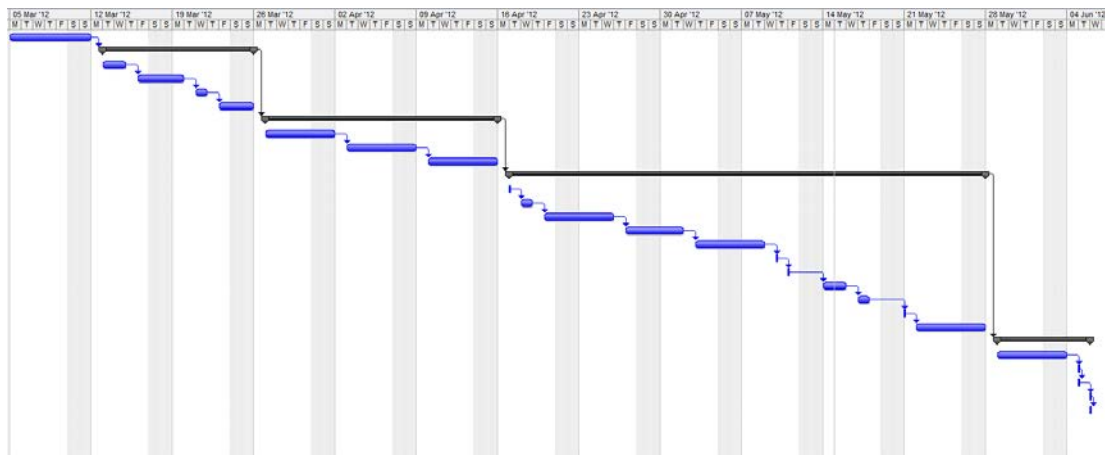
Aquestes contindran una sèrie de subtasques relacionades entre elles, i que per tal de dur a terme la tasca principal, s'hauran de completar totes les subtasques associades. El llistat de les mateixes i les dates associades a cadascuna d'elles es pot veure a continuació:

		Task Name	Duration	Start	Finish	Predecessors
1		Realització pla de treball	6 days	Mon 05/03/12	Mon 12/03/12	
2		Recollida de requisits / Documentació	10 days	Tue 13/03/12	Mon 26/03/12	1
3		Requisits de l'aplicació	3 days	Tue 13/03/12	Thu 15/03/12	
4		Documentació d'Android	3 days	Fri 16/03/12	Tue 20/03/12	3
5		Documentació OpenGL ES	2 days	Wed 21/03/12	Thu 22/03/12	4
6		Documentació programació videojocs	2 days	Fri 23/03/12	Mon 26/03/12	5
7		Anàlisi i disseny de l'aplicació	15 days	Tue 27/03/12	Mon 16/04/12	2
8		Definir requisits	5 days	Tue 27/03/12	Mon 02/04/12	
9		Documents d'anàlisi	5 days	Tue 03/04/12	Mon 09/04/12	8
10		Documents de disseny	5 days	Tue 10/04/12	Mon 16/04/12	9
11		Implementació del videojoc	30 days?	Tue 17/04/12	Mon 28/05/12	7
12		Menus del joc	1 day?	Tue 17/04/12	Tue 17/04/12	
13		IA dels enemics	2 days	Wed 18/04/12	Thu 19/04/12	12
14		Mòdul dels enemics	5 days	Fri 20/04/12	Thu 26/04/12	13
15		Mòdul de musica	4 days	Fri 27/04/12	Wed 02/05/12	14
16		Mòdul del jugador	5 days	Thu 03/05/12	Wed 09/05/12	15
17		Accés a dades xml (nivells, musica, etc)	1 day?	Thu 10/05/12	Thu 10/05/12	16
18		Interfície d'usuari	1 day?	Fri 11/05/12	Fri 11/05/12	17
19		Modul de les armes del jugador	3 days	Mon 14/05/12	Wed 16/05/12	18
20		Documentació del projecte	2 days	Thu 17/05/12	Fri 18/05/12	19
21		Joc de proves	1 day	Mon 21/05/12	Mon 21/05/12	20
22		Finalització del joc	5 days	Tue 22/05/12	Mon 28/05/12	21
23		Memòria i entrega	7 days	Tue 29/05/12	Wed 06/06/12	11
24		Escriure la memoria	5 days	Tue 29/05/12	Mon 04/06/12	
25		Preparar presentació	0,5 days	Tue 05/06/12	Tue 05/06/12	24
26		Preparar video	0,5 days	Tue 05/06/12	Tue 05/06/12	25
27		Preparar el producte per l'entrega	0,5 days	Wed 06/06/12	Wed 06/06/12	26
28		Entrega final	0,5 days	Wed 06/06/12	Wed 06/06/12	27

Planificació 2

Com es pot veure en el gràfic següent al ésser un projecte portat a terme una sola persona, no hi hauran tasques que es solapin en el temps, ni que es portin a terme de manera paral·lela, ja que aquest punt no serà possible en un projecte unipersonal. També es pot observar com les tasques estan distribuïdes en el temps (línia de color gris), i a sota d'aquesta la distribució de cadascuna de les subtasques (línia blava).

Algunes conclusions que es poden treure, es que la part dedicada a l' implementació és la més gran de totes les tasques, i potser la dedicada a la finalització de la memòria es una mica curta. Aquestes només son unes conclusions preliminars, observant aquest gràfic, més endavant tractarem el tema de la validesa de la planificació inicial, els desviaments que s'han dut a terme i els motius del mateix.



Planificació 3

Descripció de les tasques.

En aquest apartat es descriurà més en detall en que consisteixen cadascuna de les tasques i planificades, el temps que es trigarà en dur-les a terme, els objectius i fites de la mateixa.

Tasca 1:

Temporització: 1 setmana (del 5 al 11 de Març).

Descripció: Realització del document "Pla de treball":

- Estudi de la magnitud del projecte.
- Decisió de les tecnologies a utilitzar.
- Temporització del projecte.

Objectius:

- Aprofundir els coneixements sobre el desenvolupament Android.
- Tenir una visió clara sobre la funcionalitat que haurà de tenir el videojoc.
- Tenir una visió de les funcionalitats que haurà el mateix.
- Aprofundir en els coneixements de OpenGL ES

Fites:

- Document del Pla de treball.

Tasca 2:

Temporització: 2 setmanes (del 11 al 25 de Març).

Descripció: Recollida i classificació de tota la informació que pugui ser rellevant per dur a terme el projecte:

- Temes relacionats amb el desenvolupament Android.
- Temes relacionats amb el desenvolupament OpenGL ES per Android.
- Temes relacionats amb la programació de videojocs per plataformes Android

Objectius:

- Aprofundir els coneixements sobre programació Java per la plataforma Android.
- Tenir una visió clara sobre la funcionalitat que haurà de tenir el videojoc.
- Tenir una visió general sobre els videojocs més estesos / addictius del moment.

Fites:

- Document descriptiu del desenvolupament Android.
- Document d'anàlisi sobre videojocs existents.
- Document relatiu sobre API / framework per el desenvolupament de videojocs for Android.
- Document del pla de treball (entrega).

Tasca 3:

Temporització: 3 setmanes (26 de Març al 15 d'abril).

Descripció: Anàlisi i disseny del videojoc a implementar.

Objectius:

- Establir es requisits que haurà de tenir el videojoc.
- Obtenir l'anàlisi del programari en funció dels requisits i de la funcionalitat que se li vol donar.
- Obtenir el disseny de l'aplicació a implementar basant-nos en l'anàlisi del punt anterior.

Fites:

- Obtenir l'anàlisi i el disseny de l'aplicació documentant les decisions preses.
- PAC 2 del PFC

Tasca 4:

Temporització: 5 setmanes (16 d'abril al 20 de Maig).

Descripció: Implementació del videojoc.

Objectius:

- Implementar entre el 80-90% de codi font.
- Generar els jocs de proves adients per garantir el correcte funcionament.

Fites:

- Creació del videojoc.
- Creació del document amb les especificacions de videojoc.
- Document sobre la implementació (decisions, tècniques emprades, ...).
- Documentació sobre els jocs de proves.
- PAC 3 del PFC

Tasca 5:

Temporització: 1 setmana (del 21 al 27 de Maig).

Descripció: Finalització del codi i continuar amb la memòria.

Objectius:

- Finalitzar la implementació del codi.
- Continuar amb la redacció de la memòria.

Fites:

- Finalització de la fase de desenvolupament.
- Document sobre la implementació i el joc de proves.

Tasca 6:

Temporització: 1 setmana (del 28 de Maig al 3 de Juny).

Descripció: Documentació del producte.

Objectius:

- Documentar la instal·lació del videojoc.
- Documentar l'ús de les diferents opcions de configuració i / o jugabilitat del videojoc.

Fites:

- Documentació del producte creat.

Tasca 7:

Temporització: 1 setmana (del 4 al 6 de Juny).

Descripció: Arranjaments finals de la memòria i preparar la presentació del videojoc que s'ha creat.

Objectius:

- Síntesi de la memòria presa durant el projecte.
- Crear una presentació del videojoc creat.

Fites:

- Memòria del projecte.
- Presentació del videojoc.
- PAC 4 – Entrega final.

Desviaments del pla

Els desviaments en el pla de treball, han estat menors al llarg del semestre, ja que al final s'ha arribat a temps a l'entrega final. Hi ha hagut petits desviaments que passo a enumerar a continuació:

- A la PAC 2 i 3 el estat de la memòria era menor al esperat. La solució ha estat treballar mes a fons en aquest punt per assolir l'ultima entrega.
- El codi a la PAC 3 estava una mica endarrerit, i per tal d'assolir l'entrega final, s'han hagut de fer "hores extra".

En general s'han respectat la planificació inicial, amb petits desviaments al llarg del projecte, posant especial atenció a que la part de desenvolupament, ha estat molt mes extensa de lo planificat.

El context.

En aquest capítol veurem el context en el qual vaig decidir-me per aquest tipus de projecte. Veurem com el mercat dels videojocs es prou gran com per justificar aquest, i a més a més, veurem altres opcions similars al nostre producte objectiu. Així doncs primer veurem una petita introducció als videojocs, després parlaré d'altres videojocs per dispositius mòbils que hi ha al mercat i al final un petit resum de tot lo vist.

Introducció i història dels videojocs.

Introducció als videojocs

Que es un videojoc. Es sol definir com el programa informàtic que serveix per entretenir els seus usuaris. Encara que correcta aquesta definició es una mica curta, ja que, si be conte una part de lo que es un videojoc, hi ha varis aspectes que es queden en el tinter. Ja que per exemple no tots els videojocs serveixen per entretenir. Hi ha videojocs que també serveixen per produir en el usuari tot una sèrie d'emocions com poden ser la por, alegria, etc. També els hi ha que serveixen per educar. Així doncs, aquesta definició es una mica curta. Segons el diccionari Ramon Llull un videojoc es:

“Dispositiu electrònic que permet, mitjança’n els comandaments apropiats, simular jocs en las pantalles d’un ordinador o la televisió”.

Així doncs, es defineix com “un exercici recreatiu, amb regles en el qual es pot guanyar o perdre”. Però això tampoc es 100% correcte ja que hi ha videojocs que es la finalitat no es guanyar com pot ser el Simcity or els Sims.

Per tant , videojoc es aquell programa informàtic, normalment associat a un hardware específic, que recrea un exercici sotmès a regles, per aconseguir un o varis objectius, on els jugadors podran interactuar i prendre decisions.

Història

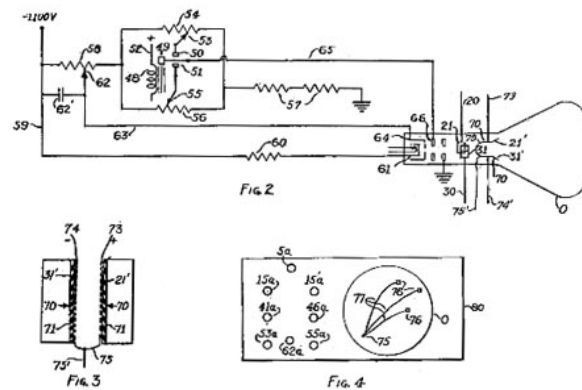
L'inici dels videojocs es una mica difús, i depèn es clar de la nostra idea de videojoc, per dir quin va ser el primer de l' història. Per un canto hi ha varis autors que creuen que les primeres maquines recreatives com el pinball son videojocs pròpiament dits, i uns altres entenen per videojocs aquells que fan servir un dispositiu de vídeo.

Com es podrà veure l'història dels videojocs evoluciona paral·lelament a la història del hardware, ja que estan íntimament relacionats. A millor hardware, millors jocs podrem desenvolupar. En qualsevol cas farem una distinció entre dues etapes de la història dels videojocs:

Primera etapa

Comença a finals dels anys cinquanta quan es conceptualitza el videojoc. En aquesta etapa es desenvolupen varis projectes d'investigació que al cap i a la fi seran com un divertiment per els seus autors. Com a exemples d'aquesta primera etapa tenim el famós Cathode-Ray Tube Amusement Devide. Es va crear el 14 de desembre del 1948 per Thomas T. Goldsmith, Cedar Grove i Estle Ray. Va ser presentada a l'oficina de patents dels EEUU amb numero 2.455.922. Aquest petit videojoc simulava el llançament de míssils òbviament fent servir l'electrònica de l'època.

Esquema presentat a la patent.



Context 1

I uns petits exemples com el tres en ratlla, desenvolupat en 1952 per el estudiant Alexander Douglas, com a tesis doctoral de la universitat de Cambridge, i Space War desenvolupat alt 1962 per Steve Russell, Martin Graetzky i Wayne Wiitanen, per a la computadora DEC PDP-1.



Context 2

Segona etapa

Arrenca a principis dels anys 80, quan es crea un nou mercat dels videojocs amb l'arribada de les maquines recreatives del tipus arcade. A partir d'aquest moment es crea una gran

industria del videojoc arribant als nostres dies on ja supera en facturació a l'industria audiovisual. Com exemples de l'industria audiovisual de finals dels 80 i començament del 90 tenim Super Mario Bros o el Tomb Raider.



Context 3

El mercat del videojoc

Els videojocs estan dintre del segment del oci. Tradicionalment dintre d'aquest segment hi era la musica, el cinema, els llibres, teatre, espectacles, etc. Amb l'aparició dels videojocs i el pas dels anys, aquest sector ha sofert canvis, i ara el teatre per exemple no hi es dintre del segment del oci, i si els videojocs. Els ingressos per videojocs, han augmentat fins al punt de a l'any 2006 segons una estadística presentada per la ADESE (Associació Espanyola de Distribuïdors i Editors de Software d'Entreteniment), va suposar mes que el cinema i la musica junts.

Segment	Consum M€	Percentatge consum
Pel·lícules Vídeo	442 M€	18,3%
Musica Gravada	372 M€	15,4%
Taquilla Cinema	636 M€	26,3%
Videojocs	967 M€	40%

Aquesta estadística es pot desglossar el apartat de videojocs amb un 26% software i un 14% hardware. Així doncs veiem que gairebé la meitat del mercat del oci esta ocupat per l'industria dels videojocs. Per altra banda d'aquesta estadística també es desprenen dades interessants com que els videojocs nomes ocupen el 9% de temps d'oci dels espanyols, o que la majoria dels usuaris 71% prefereix jugar en grup.

La cultura dels videojocs

Al voltant dels videojocs ha nascut una cultura pròpia que ha evolucionat paral·lelament al mercat. Per alguns els videojocs es gairebé una forma de vida, arribant al extrem que actualment hi existeixen feines en les quals et paguen nomes per jugar. També hi ha països

com Korea on es retransmetien campionats de videojocs en Prime-Time per televisió, o els EEUU on hi han jugadors patrocinats per diferents companyies.

Un exemple de les competicions a nivell mundial son les World Series of Videogames, que es un circuit professional per a jugadors de videojocs, que celebra varis tornejos anualment, per proclamar els millors jugadors i equips dels jocs mes famosos del moment (com poden ser Quake, Counter-Strike, WoW,....).

Un altre exemple es l'emissora de Videojocs GamePro TV la qual es dedica única i exclusivament a la retransmissió d'esdeveniments relacionats amb els videojocs. Per altre banda tenim videojocs que tenen una adaptació al cinema com pot ser Resident Evil, Tomb Raider, Max Payne entre d'altres.

Després de aquesta petita introducció als videojocs, la seva historia, el segment que representen en el mercat i el moviment cultural que hi ha al voltant podem deduir que:

- Es una industria jove, amb nomes 20 anys de vida, que canvia constantment i que en els propers anys, amb d'implantació de tecnologies com l' ADSL, el component en xarxa serà molt important.
- Actualment representa gairebé la meitat de tota l'industria del oci amb una facturació a Espanya d' aproximadament de 1000 milions d' euros.
- Que hi ha tot un moviment cultural al darrera on fins i tot hi ha gent que cobrar per jugar.
- Que hi ha tota una comunitat de desenvolupament darrera just pel plaer de desenvolupar o com a hobby, que reclamen eines per desenvolupar de manera gratuïta, ja que no hi ha interès comercial, nomes personal.

Així doncs, com a resum, hem vist que els videojocs es un sector encara jove, amb molt per créixer, i que hi ha tota una comunitat de desenvolupadors amateurs darrera.

Android

Android és un sistema operatiu mòbil basat en Linux, que junt amb aplicacions "middleware" està enfocat per ser utilitzat en dispositius mòbils com telèfons intel·ligents, tabletetes, Google TV i d'altres dispositius. És desenvolupat per la Open Handset Alliance, la qual està liderada per Google.

Va ser desenvolupat inicialment per Android Inc., una petita companyia amb seu a Palo Alto (Califòrnia) fundada al Octubre del 2003. Molts dels fundadors van passar a formar part després del equip de desenvolupament a Google.

Android Inc. Va passar gairebé dos anys treballant en la ombra, fins que Google va adquirir la companyia en Agost del 2005. Però la data clau en l'història d'Android va ser el 5 de Novembre de 2007, quan va ser fundada OHA (Open Handset Alliance), que era una aliança comercial de 35 companyies liderades per Google, entre els que es trobaven fabricants de mòbils, operadors de telecomunicacions, fabricants de chips i desenvolupadors de programari. Aquell mateix dia es va anunciar el sistema Android, una plataforma de codi obert per mòbils, que presentava com a novetat que estava basada en Linux. Pocs dies després, el 12 de Novembre, Google va alliberar el primer kit de desenvolupament d'aplicacions, sota llicència Apache, i de codi obert.

En l'actualitat és el sistema operatiu més utilitzat en dispositius mòbils. La principal raó per la qual Android ha esdevingut el sistema més popular entre aquest tipus de dispositius es el fet que, a diferència de la resta (IOS, Blackberry OS, Windows phone) el seu codi és obert, lo que significa que els desenvolupadors poden accedir al codi font, on es poden reportar problemes previs i corregir-los, existint una enorme comunitat d'usuaris que donen el seu suport per la millora del sistema operatiu Android.

Historia d'Android

En aquest apartat, parlarem breument de l'història i els canvis més transcendents d'aquest sistema operatiu per dispositius mòbils:

- Android 1.0. (23 Setembre de 2008), Apareix el primer dispositiu mòbil amb Android, el HTC Dream.
- Android 1.1 (8 de Febrer de 2009), Primera actualització del sistema operatiu.
- Android 1.5 Cupcake (30 Abril 2009), Aquesta nova versió va incloure com a millores:
 - Major rapidesa de la càmera i captura d'imatges.
 - Millores en la localització per GPS.
 - Teclat tàctil en pantalla.
 - Pujada directa de vídeos i fotos a Youtube i Picassa, respectivament.
- Android 1.6 Donut (15 de Setembre de 2009), Aquesta nova versió va incloure com a millores:
 - Reconeixement de veu,
 - Millora de la captura d'imatges i vídeo.
 - Indicador del estat de la bateria.
 - Suport CDMA.
 - Funció Veu-Text multi llenguatge.

- Android 2.0 Eclair (25 Octubre 2009):
 - Sincronització de múltiples comptes de correu.
 - Suport per Microsoft Exchange.
 - Suport per Bluetooth 2.1
 - Actualització del navegador per suport HTML5.
 - Noves funcions de calendari.
- Android 2.01 i 2.1 no van incloure millores rellevants.
- Android 2.2 Froyo:
 - Nous Widgers per configurar el dispositiu.
 - Millorat el suport per Exchange.
 - Suport per Hotspot.
 - Teclat multi llenguatge.
 - Adobe Flash 10.1
- Android 2.3 Gingerbread (6 Desembre de 2010):
 - Millora en la interfície d'usuari.
 - Nou teclat per pantalla millorat.
 - Near Field Communication (NFC).
 - Internet celling.
- Les versions 2.3.3 i 3.0 no van aportat grans novetats.
- Android 3.1 Honeycom (10 de Maig de 2011):
 - Optimitzat per tabletas i dispositius de pantalla gran.
 - Millora en la multitasca, notificaciones, i els Widgets.
 - Bluetooth tethering.
 - Suport natiu per la transferència de medis.
- Android 3.2 no va aportat grans novetats.
- Android 4.0 Ice Cream Sandwich (19 d'Octubre de 2011):
 - Nou estil d'escriptura
 - Funció de reconeixement facial.
 - Nou "framework" unificat per aplicacions.
 - Suport natiu per el us de ploma.

Com hem vist durant el llistat anterior, les versions d'Android més rellevants rebran noms de postres (en anglès). Per cada nova versió es tria un postra que comenci amb la següent lletra del abecedari:

Lletra / Versió	Postra
C: Cupcake (v1.5)	Magdalena glacejada
D: Donut (v1.6)	Rosquilla
E: Éclair (v2.0/v2.1)	Pastel francès
F: Froyo (v2.2)	(abreviatura de «frozen yogurt») logurt gelat
G: Gingerbread (v2.3)	Pan de jengibre
H: Honeycomb (v3.0/v3.1/v3.2)	Panal
I: Ice Cream Sandwich (v4.0)	Sandwich de gelat
J: Jelly Bean (v5.0)	Gominoles

Google Play

Les aplicacions es desenvolupen habitualment en el llenguatge Java amb el ADT (Android Development Tools), però es poden trobar d'altres eines de desenvolupament. En número d'aplicacions va arribar fa poc a la xifra de 400.000 (2/3 de les qual son gratuïtes) i als 10.000.000 de descarregues (motiu pel qual, Google va llençar una agressiva campanya publicitària).

Google Play és la botiga en línia de programes i videojocs desenvolupada per Google per dispositius Android. L'aplicació s'anomena "Play Store" i es troba instal·lada a la majoria de dispositius Android, i permet els seus usuaris navegar i descarregar aplicacions publicades per la comunitat de desenvolupadors. Google retribueix als desenvolupadors el 70% del preu de les aplicacions. Google Play no es l'única botiga virtual, ja que existeixen d'altres com Amazon AppsStore o SlideME. Un altre opció es instal·lar els binaris directament al dispositiu mòbil.

Per últim, veurem una llista dels 10 jocs que son súper vendes i una petita descripció dels mateixos, per entendre millor com està el mercat de videojocs a Google Play:

- **Assassin's Creed Altair's Chronicles:** Joc d'acció en primer persona, consisteix en la preqüela del primer Assassin's Creed, que ens situa en l'època de les Creuades on controlem Altair en el seu intent de recuperar el sant grial. [Enllaç a Google Play](#)
- **Hero of Sparta:** Joc d'acció en primera persona on controlarem a un espartà que s'enfrontarà a diversos enemics, amb el seu periple per poder tornar al seu hogar. [Enllaç a la web de GameLoft](#)
- **Modern Combat 2:** Joc d'acció en primera persona, on tomarem el control d'un oficial del exercit i haurem de combatre contra terroristes. Una gran adaptació de la seva versió en PC. [Enllaç a Google Play](#)

- **Cut the Rope:** Joc arcade, el objectiu de cada nivell del joc es maniobrar un tros de caramel fins a la boca d'un monstre anomenat Om Nom. [Enllaç a Google Play](#)
- **Fruit Ninja:** Joc arcade/casual on el jugador lliscarà els seus dits per la pantalla, amb el objectiu de partir en trossos la fruita que apareix per pantalla. Joc de temàtica molt simple, però amb una gran acceptació per la seva addicció. [Enllaç a Google Play](#)
- **Asphalt 6:** Simulador esportiu sobre carreres de cotxes, amb la possibilitat de jugar en local y remot. [Enllaç a Google Play](#)
- **Plants vs. Zombies:** Joc d'estratègia, on col·locarem distint tipus de plantes amb diferents característiques per tal de defensar-nos d'una horda de zombis. [Enllaç a Google Play](#)
- **Zenonia:** RPG de temps real. El personatge principal te la missió de trobar la raó de la mort del seu pare a mans d'un dimoni. El desenvolupament dependrà de les decisions que anem prenem durant el desenvolupament del joc. Aquest seria el més conegut d'aquesta temàtica per Android. [Enllaç a Google Play](#)
- **Robo Defense:** Similar al "Plants vs. Zombies", només que en comptes de plantes construirem torretes defensives per defensar-nos de les onades d'enemics. [Enllaç a Google Play](#)
- **Angry Birds:** Joc arcade, que consisteix en controlar un grup d'ocells en la seva missió per recuperar els ous del seu niu. Pot semblar un joc molt senzill, però ha aconseguit esser tot un èxit de vendes, per això mateix, per la seva simplicitat. [Enllaç a la Play Store](#) .

Anàlisi del sistema

En aquesta secció analitzarem la fase de recollida de requisits del sistema. Per dur a terme aquesta tasca, partirem d'una vista general dels casos d'us, i a continuació detallarem cadascun dels casos, es formalitzaran i concretaran els requisits no funcional de cadascun dels requisits.

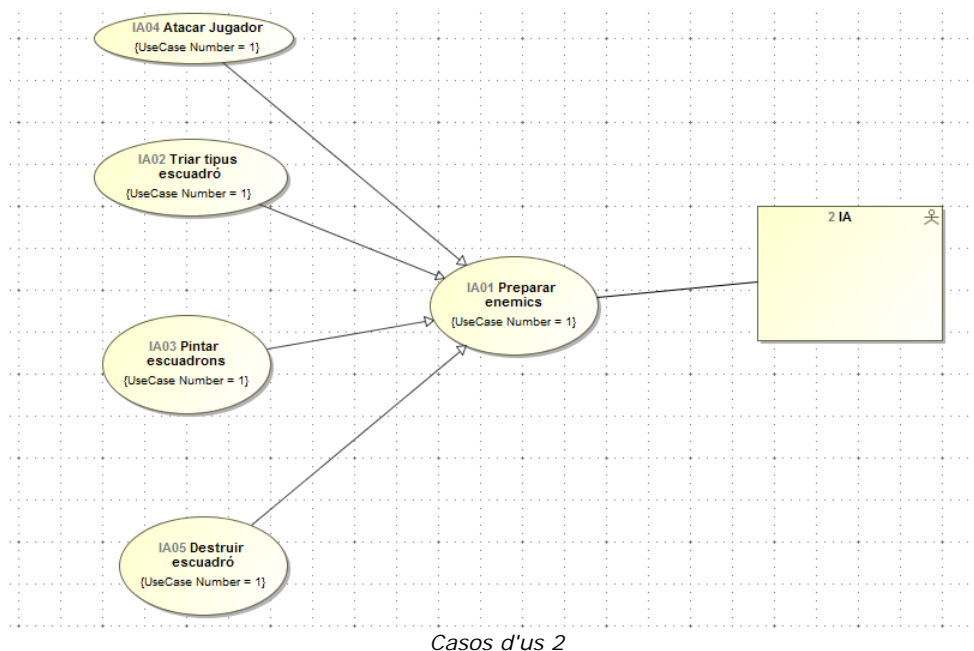
Diagrama de casos d'us

El següent diagrama recull una vista global del actor "jugador" i els casos d'us que descriueixen els requisits funcionals de l'aplicació:



Com es pot observar , el sistema distingirà 1 usuari per els casos d'us anteriors, el usuari que representarà al jugador, que serà el que interactuarà amb el sistema. Aquest usuari, com es veu en el gràfic anterior, interactuarà accedint i sortint del sistema, modificant les diferents opcions del mateix, o jugant al videojoc.

Aquest sistema també disposarà d'un actor amb una sèrie "d'intel·ligència" que seria la intel·ligència artificial del sistema, que podem veure a continuació també interactuarà amb el sistema, i per tant, serà susceptible d'analitzar el seu "us" de l'aplicació:



A mode de resum, a la següent taula es poden observar una llista de tots els casos d'us que desenvoluparem amb més detall en el següent punt.

Codi	Descripció	Actor
JU01	Iniciar el videojoc	Jugador
JU02	Sortir del videojoc	Jugador
JU03	Accedir a les opcions del videojoc	Jugador
JU04	Modificar so (SI/NO)	Jugador
JU05	Modificar vibració (SI/NO)	Jugador
JU06	Modificar nivell de dificultat	Jugador
JU07	Jugar	Jugador
PN01	Iniciar partida nova al videojoc	Jugador
PN02	Finalitzar partida	Jugador
PN03	Jugar partida	Jugador
IA01	Preparar esquadrons d'enemics	IA
IA02	Pintar esquadrons	IA
IA03	Atacar jugador depenent del tipus d'esquadró	IA
IA04	Destruir escuadró	IA

Descripció dels casos d'us

En aquest punt descriurem en detall cadascun dels casos d'us que s'han detectat en el apartat anterior, per tal de fer un estudi més en profunditat de cadascun d'ells.

JU01 Iniciar Videojoc

Identificador	JU01
Nom	Iniciar Videojoc
Autor	Alejandro Santiago
Resum	El usuari inicia el videojoc en el seu dispositiu Android
Actor	Jugador
Precondicions	Cap. El programari encara no estava actiu i per tant no pot tenir cap precondició
Postcondicions	El usuari ha inicialitzat el videojoc de manera satisfactòria, i un cop vista la presentació es troba en la pantalla del menú principal.
Flux normal	<ol style="list-style-type: none"> 1. El usuari accedeix al llistat d'aplicacions del seu dispositiu 2. Un cop al llistat, busca el accés directe del videojoc 3. Executa el programari 4. Es comença amb el procés de carrega 5. El videojoc mostra la pantalla de benvinguda i la presentació 6. Es mostrarà el menú principal al usuari
Flux alternatiu	<p>Es podria donar el cas que el usuari disposi d'un accés directe del videojoc en algun dels escriptoris del qual es disposa en els dispositius d'aquest tipus.</p> <ol style="list-style-type: none"> 1. El usuari accedeix al escriptori on te el accés directe al videojoc 2. Executa el programari 3. Es comença amb el procés de carrega 4. El videojoc mostra la pantalla de benvinguda i la presentació 5. Es mostrarà el menú principal al usuari
Inclusions	Cap
Exclusions	Cap

JU02 Sortir del videojoc

Identificador	JU02
Nom	Sortir del videojoc
Autor	Alejandro Santiago
Resum	El usuari decideix sortir del videojoc
Actor	Jugador
Precondicions	El usuari ha iniciat el programa i es troba en el menú principal
Postcondicions	Un cop tancada la aplicació el usuari es troba de tornada al escriptori o al menú, depèn d'on es trobés abans d'iniciar la aplicació
Flux normal	<ol style="list-style-type: none"> 1. El usuari ha d'anar al menú principal 2. El videojoc mostrarà el menú principal 3. Seleccionar la opció de "EXIT" 4. El programari es descarrega de memòria i surt
Flux alternatiu	<p>Es podria donar el cas que el usuari utilitzi el botó del que disposen tots els dispositius Android, el de tornar endarrere. En aquest cas també es pot sortir del nostre programari</p> <ol style="list-style-type: none"> 1. En qualsevol punt del videojoc, el usuari pot prémer aquest botó

Inclusions	2. Depenen del punt on es trobi, l'haurà de prémer tantes vegades com sigui necessari fins sortir de l'aplicació
Exclusions	3. El programari es descarrega de memòria i surt
	JU01
	Cap

JU03 Accedir a les opcions del videojoc

Identificador	JU03
Nom	Accedir a les opcions del videojoc
Autor	Alejandro Santiago
Resum	El usuari vol accedir a les opcions del videojoc
Actor	Jugador
Precondicions	El usuari ha iniciat el programa i es troba en el menú principal
Postcondicions	El usuari triarà quina de les opcions vol modificar, o tornar
Flux normal	<ol style="list-style-type: none"> 1. El usuari accedeix al menú, opcions 2. El videojoc carrega i mostra al usuari el menú d'opcions
Flux alternatiu	Cap
Inclusions	JU01
Exclusions	Cap

JU04 Modificar so (SI/NO)

Identificador	JU04
Nom	Modificar so (SI/NO)
Autor	Alejandro Santiago
Resum	El usuari del videojoc, voldrà seleccionar si desitja jugar amb so ambient o sense
Actor	Jugador
Precondicions	El usuari ha iniciat el programa i es troba dintre del menú de les opcions
Postcondicions	El usuari triarà si vol jugar amb so o sense
Flux normal	<ol style="list-style-type: none"> 1. El usuari selecciona la opció de so en el menú d'opcions 2. El programa mostrarà el menú d'opcions de so 3. El usuari seleccionarà si desitja so o sense so 4. El videojoc tornarà al menú principal
Flux alternatiu	<p>Tots els dispositius mòbils tenen una manera de apujar el volum o baixar-lo, per tant es podria abaixar el volum del joc fins arribar al silenci, o viceversa</p> <ol style="list-style-type: none"> 1. El usuari decideix baixar el volum del joc 2. Pressiona el botó de baixar el volum 3. El dispositiu Android baixa el volum, i per tant també del videojoc
Inclusions	JU03
Exclusions	Cap

JU05 Modificar vibració (SI/NO)

Identificador	JU05
Nom	Modificar vibració (SI/NO)
Autor	Alejandro Santiago
Resum	El usuari del videojoc, voldrà seleccionar si desitja jugar amb vibració o sense
Actor	Jugador
Precondicions	El usuari ha iniciat el programa i es troba dintre del menú de les opcions
Postcondicions	El usuari triarà si vol jugar amb so o sense
Flux normal	<ol style="list-style-type: none"> 5. El usuari selecciona la opció de vibració en el menú d'opcions 6. El programa mostrarà el menú d'opcions de vibració 7. El usuari seleccionarà si desitja so o sense vibració 8. El videojoc tornarà al menú principal
Flux alternatiu	No hi ha
Inclusions	JU03
Exclusions	Cap

JU06 Modificar nivell de dificultat

Identificador	JU06
Nom	Modificar nivell de dificultat
Autor	Alejandro Santiago
Resum	Modificar el nivell de dificultat del videojoc
Actor	Jugador
Precondicions	El usuari ha iniciat el programa i es troba dintre del menú de les opcions
Postcondicions	El usuari triarà el nivell de dificultat amb la que vol jugar al videojoc
Flux normal	<ol style="list-style-type: none"> 1. El usuari selecciona la opció de dificultat en el menú d'opcions 2. El programa mostrarà el menú d'opcions de dificultat 3. El usuari seleccionarà el nivell que desitja de dificultat 4. El videojoc tornarà al menú principal
Flux alternatiu	cap
Inclusions	JU03
Exclusions	No hi ha

JU07 Jugar

Identificador	JU07
Nom	Jugar
Autor	Alejandro Santiago
Resum	Jugar al videojoc
Actor	Jugador
Precondicions	El usuari ha iniciat el programa i es troba dintre del menú principal
Postcondicions	El jugador finalitza la partida, o surt del programa
Flux normal	<ol style="list-style-type: none"> 1. El jugador selecciona la opció jugar en el menú principal 2. Comença el joc, i el jugador pot interactuar amb el mateix 3. El joc termina i torna al menú principal
Flux alternatiu	<ol style="list-style-type: none"> 1. Un cop comença el joc, i el jugador pot interactuar amb el mateix 2. En qualsevol moment, el usuari decideix sortir al menú principal
Inclusions	JU01
Exclusions	JU03

PN01

Identificador	PN01
Nom	Iniciar partida nova
Autor	Alejandro Santiago
Resum	El jugador vol iniciar una partida nova en el videojoc
Actor	Jugador
Precondicions	El jugador ha iniciat el videojoc en el seu dispositiu
Postcondicions	Cap
Flux normal	<ol style="list-style-type: none"> 1. El jugador fa clic en la opció de jugar en el menú principal 2. Comença a executar-se la partida
Flux alternatiu	No hi ha
Inclusions	JU01
Exclusions	JU03

PN02

Identificador	PN02
Nom	Finalitzar partida
Autor	Alejandro Santiago
Resum	El jugador vol donar per finalitzada la par
Actor	Jugador
Precondicions	El jugador estava jugant una partida
Postcondicions	Cap
Flux normal	<ol style="list-style-type: none"> 1. El jugador esta jugant una partida 2. Pressiona el boto de tornar enrere en el seu dispositiu mobil
Flux alternatiu	Cap
Inclusions	JU01, PN01
Exclusions	JU03

PN03

Identificador	PN03
Nom	Jugar partida
Autor	Alejandro Santiago
Resum	El jugador juga una partida en el videojoc
Actor	Jugador
Precondicions	El jugador ha iniciat el videojoc, i ha decidit iniciar la partida
Postcondicions	Cap
Flux normal	<ol style="list-style-type: none"> 1. El jugador comença la partida 2. Amb el sistema d'entrada i sortida decideix com jugar 3. En cas de desitjar canviar qualsevol opció pot sortir al menú principal
Flux alternatiu	Cap
Inclusions	JU01, PN01
Exclusions	PN02, JU03

IA01

Identificador	IA01
Nom	Preparar esquadrons d'enemics
Autor	Alejandro Santiago
Resum	La IA del videojoc prepara un nou esquadró d'enemics
Actor	Intel·ligència Artificial del videojoc
Precondicions	El jugador ha iniciat una partida
Postcondicions	Començarà a moure el esquadró
Flux normal	<ol style="list-style-type: none"> 1. El programa selecciona el tipus d'esquadró 2. Defineix la posició, composició i estat 3. Posiciona el esquadró a la pantalla i el activa
Flux alternatiu	Cap
Inclusions	JU01, PN01
Exclusions	PN02, JU03

IA02

Identificador	IA03
Nom	Pintar esquadrons
Autor	Alejandro Santiago
Resum	Es pintaran els esquadrons visibles a la pantalla del videojoc
Actor	Intel·ligència Artificial del videojoc
Precondicions	El jugador esta jugant, i el nivell encara no ha arribat al final
Postcondicions	Cap
Flux normal	<ol style="list-style-type: none"> 1. La IA recórrer tots els esquadrons 2. Comprova si estan visibles a la pantalla 3. En cas afirmatiu, els pinta per pantalla
Flux alternatiu	Cap
Inclusions	JU01, PN01, IA01, IA02
Exclusions	PN02, JU03

IA04

Identificador	IA04
Nom	Atacar jugador
Autor	Alejandro Santiago
Resum	Atacar jugador depenent del tipus d'esquadró
Actor	Intel·ligència Artificial del videojoc
Precondicions	El jugador esta jugant, i el nivell encara no ha arribat al final
Postcondicions	Cap
Flux normal	<ol style="list-style-type: none"> 1. La IA recórrer tots els elements del esquadró 2. En cas que el enemic, sigui del tipus que dispara 3. "Sortejarà" si ha de disparar (20% de probabilitats) 4. En cas que s'hagi de disparar, crearà l'arma enemiga
Flux alternatiu	<ol style="list-style-type: none"> 1. Hi ha tipus d'esquadrans que persegueixen el jugador 2. En cas que sigui el tipus, modificarem el moviment del esquadró per atacar
Inclusions	JU01, PN01, IA01, IA02
Exclusions	PN02, JU03

IA05

Identificador	IA05
Nom	Destruir esquadró
Autor	Alejandro Santiago
Resum	El esquadró ha estat destruït o ha sortit per la part inferior de la pantalla
Actor	Intel·ligència Artificial del videojoc
Precondicions	El jugador esta jugant, i el nivell encara no ha arribat al final
Postcondicions	Cap
Flux normal	<ol style="list-style-type: none"> 1. El esquadró esta a la part visible de la pantalla 2. El jugador destrueix el últim component del esquadró 3. La IA assigna el estat de destruït al esquadró
Flux alternatiu	<ol style="list-style-type: none"> 1. El esquadró amb algun dels seus components vius, arriba al final de la pantalla 2. La IA detecta el punt anterior, i el destrueix, ja que ha desaparegut.
Inclusions	JU01, PN01, IA01, IA02
Exclusions	PN02, JU03

Així doncs com a resum dels casos d'us, hem vist que hi ha dos tipus d'actors, un obvi que serà el usuari o jugador, amb les típiques accions de qualsevol jugador de videojocs, i la intel·ligència artificial que manegarà tot lo referent als enemics, en cas que el jugador decideix jugar una partida en el nostre videojoc

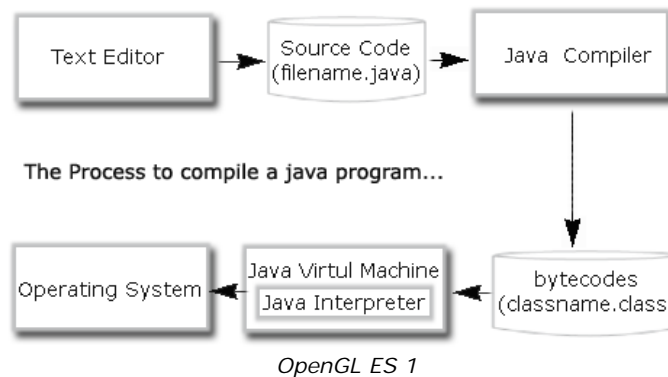
Descripció de les tecnologies utilitzades

Java

Va ser desenvolupat per James Gosling i els seus companys de Sun Microsystems a començaments de la dècada dels 90. El objectiu era utilitzar-lo com a “set-top box”, un tipus de dispositiu que s’encarrega de la recepció i descodificació de senyal televisiva. El seu nom originalment va ser Oak, més tar va ser conegut com a Green, i finalment com a Java en honor a la illa Indonèsia de Java on és van reunir els seus creadors durant el procés de desenvolupament del mateix. El objectiu original de Sun era el de crear un llenguatge amb una estructura i sintaxis similar al C i C++, encara que simplificant el model d’objectes i eliminat totes les eines de baix nivell.

Els cinc pilars de Java won: la programació orientada a objectes, la possibilitat d’utilitzar el mateix programa en diversos sistemes operatius, la inclusió del suport per xarxa per defecte (Sockets), la opció d’executar codi en sistemes remots (RMI) i la seva facilitat d’us.

El procés de compilació del llenguatge Java introdueix algunes novetats, com la generació d’un codi intermedi que serà interpretat per una maquina virtual, on s’executarà aquest codi intermedi o “bytecode”, i no en el sistema operatiu pròpiament dit, com es pot observar en el següent gràfic:



Les aplicacions de Java son molt amplies, ja que s'utilitza en una gran varietat de dispositius mòbils, petits electrodomèstics, aplicacions web, d'escriptori i moltes més.

ADT (Android Development tools)

Google proveeix del Android Development Tools (ADT) com a “framework” de desenvolupament d’aplicacions per Android, juntament amb Eclipse que es el IDE suportat oficialment. ADT conté totes les funcionalitats requerides, per crear, compilar i desplegar aplicacions per Android, utilitzant tant Eclipse com la línia de comandes. Hi ha adaptacions per poder fer servir altres IDEs com IntelliJ o Netbeans, però no estan suportats oficialment. Per últim ADT també proveeix d’un emulador, amb el qual es poden provar les aplicacions Android sense un dispositiu Android real.

Les aplicacions Android estan escrites principalment en Java, les quals seran compilades i convertides en *.class per el JDK. Un cop això, el ADT proveeix d'una eina anomenada "DX" que converteix aquestes classes Java en fitxers *.dex. Durant aquest procés de conversió s'optimitza la classes i s'emmagatzemen totes dintre d'un únic fitxer, i per tant tindrà una mida molt més petita que el conjunt de classes original.

Un cop realitzat aquest pas, aquest fitxer *.dex i tots els recursos de l'aplicació (imatges, fitxers XML, música, etc), s'empaquetaran dintre d'un fitxer *.apk que serà el instal·lable o binari del nostre projecte. Aquest fitxer resultant doncs, es podrà instal·lar en dispositius Android via la eina "adb" (manualment) o a través de Google Play.

OpenGL ES

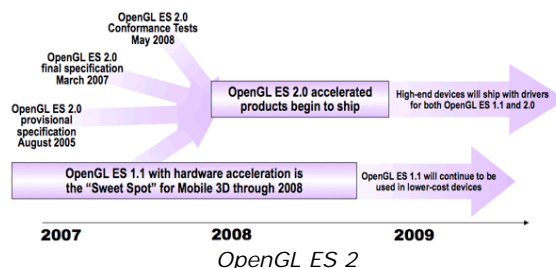
OpenGL ES és un acrònim per OpenGL for Embedded Systems, el qual es una plataforma lliure per el manegament de gràfics en 2D i 3D en sistemes encastats ("Embedded Systems"). Consisteix en un subconjunt de les rutines de OpenGL per PC, creant una interfície a baix nivell entre el programari i el hardware d'acceleració gràfica molt potent. Les característiques principals de les dues versions existents de OpenGL ES son les següents:

- OpenGL ES 1.x: (per maquinari fix) Està definida seguint la versió 1.5 de OpenGL i posa èmfasi en l'acceleració per maquinari a part de ser totalment compatible amb la versió 1.0. Proveeix una millor funcionalitat, qualitat d'imatge i optimització per augmentar el rendiment i reduir el us de memòria per estalviar energia (punt crític en els dispositius mòbils).
- OpenGL ES 2.0: (per maquinari programable), Està definida seguint la versió 2.0 de OpenGL, posa accent a la part de programació gràfica en 3D amb la capacitat de crear "shader", "vèrtex", ombres etc (característiques útils a l'hora de fer els càlculs d'aquests objectes).

La evolució en el temps de OpenGL ES" és fa de manera paral·lela per les dues versions (1.X i 2.0) i actualment la majoria de dispositius donen suport per les dues:

OpenGL ES Evolution

- **OpenGL ES 2.0 silicon implementations now shipping**
 - Shader-based graphics comes to mobile
 - Conformance tests shipping in May 2008
- **Listening carefully to implementation and developer feedback**
 - The determine next-generation requirements

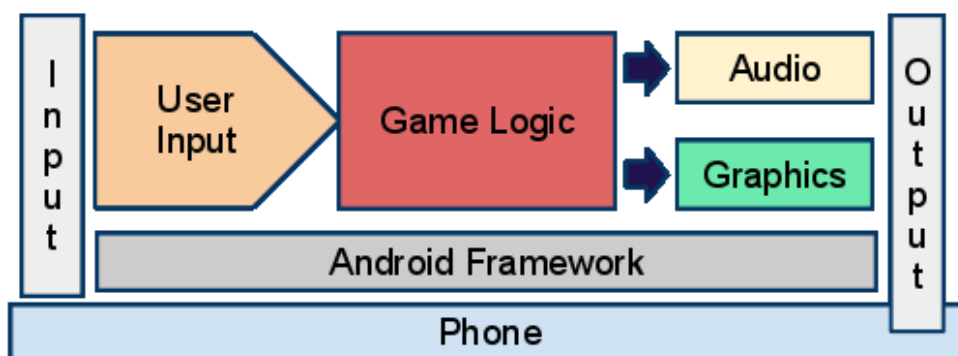


Disseny

En aquest apartat veurem tot lo relacionat amb el disseny de la nostra aplicació, videojoc en aquest cas. Parlarem de l'arquitectura, el flux del videojoc i les classes que formaran el mateix.

Arquitectura global

Els videojocs per a dispositius mòbils, Android en el cas d'aquest projecte tindran normalment una arquitectura com es pot veure en el gràfic següent:



Arquitectura 1

En el esquema anterior podem veure com primer torbem la capa del sistema operatiu del nostre telèfon, Android OS en el cas que ens ocupa. A continuació descriurem la resta de components del videojoc:

- **“Input”**: En aquest nivell trobarem la pantalla tàctil, o el teclat del telèfon, el micròfon, la càmera, els acceleròmetres, etc.
- **“User input”**: En el nostre videojoc, les entrades del usuari, serà quan pressioni la pantalla tàctil del dispositiu, a les àrees que definirem com els controls del videojoc, o els diferents botons dels diferents menús.
- **“Game Logic”**: El mòdul de la lògica del nostre videojoc, contindrà tot lo referent al control dels diferents estats del jugador, els enemics, les bales disparades per ambdós, moure el fons de pantalla, controlar el estat del jugador i qualsevol altre component “logic” del videojoc.

- **“Audio”**: Aquest mòdul produirà els sons adients per cada estat del videojoc, o per cada esdeveniment que així ho necessiti. També reproduirà la musica de fons durant els diferents nivells.
- **“Graphics”**: Aquest és el mòdul responsable de mostrar per pantalla els diferents estats del videojoc. Això pot incloure des de pintar un sol triangle, com a complexes escenes en 3D. En el cas del nostre projecte, consistirà en pintar per pantalla la nau del jugador, la dels enemics, la interfície d’usuari, el fons de pantalla.
- **“Output”**: La sortida o output seria el conjunt de la sortida dels mòduls d’àudio i de gràfics (potser també la vibració del telèfon si així ho hem decidit).

Decisions tecnològiques

A continuació detallaré les decisions tecnològiques que s’han dut a terme per el desenvolupament del videojoc, i els motius que las van motiva. Algunes com es podrà veure venen obligades per el tipus de plataforma triada per el desenvolupament del projecte:

- **Llenguatge de programació**: El llenguatge triat serà Java en la seva versió 7 (l’ultima disponible). Aquesta tria be donada per Android, i el seu SDK que obliga al desenvolupament usant Java.
- **Framework de desenvolupament**: Si es vol desenvolupar per Android, i no fer-ho a molt baix nivell, s’ha de fer servir el ADT (Android Development Tools), i per tant no hi havia cap opció si es volia treballar amb aquesta plataforma.
- **Gràfics**: Vaig decidir fer servir de manera nativa OpenGL ES en comptes de qualsevol de les llibreries gràfiques que hi ha al mercat, algunes d’elles com COCOS2D multi plataforma per dos motius principalment:
 - El meu coneixement d ‘OpenGL per la meva experiència prèvia va fer que la corba d’aprenentatge sigues molt mes rapida, per la seva similitud amb OpenGL ES
 - Amb aquesta decisió crec que el meu enteniment de la plataforma Android després del projecte es més gran per el fet de treballar mes a baix nivell que usant algun framework gràfic.

Diagrama de classes

A continuació comentarem l'estructura de les diferents classes dintre del codi font, i quina ha estat la lògica a l'hora d'agrupar aquestes. Es podrà observar una petita explicació per cadascun dels "packages" del projecte, per tal d'entendre millor el contingut dels mateixos. Per altra banda no s'ha inclòs un diagrama de classes a molt baix nivell, ja que per motius d'espai això no es possible.

Dades

Algunes classes necessitem manegar un conjunt de dades per el seu funcionament, com pot ser la classe del jugador, que haurà de controlar els punts, els escuts de la nau, etc. Com es pot veure en el diagrama a continuació no son gaire complexes, i funcionen més com un conjunt de dades:

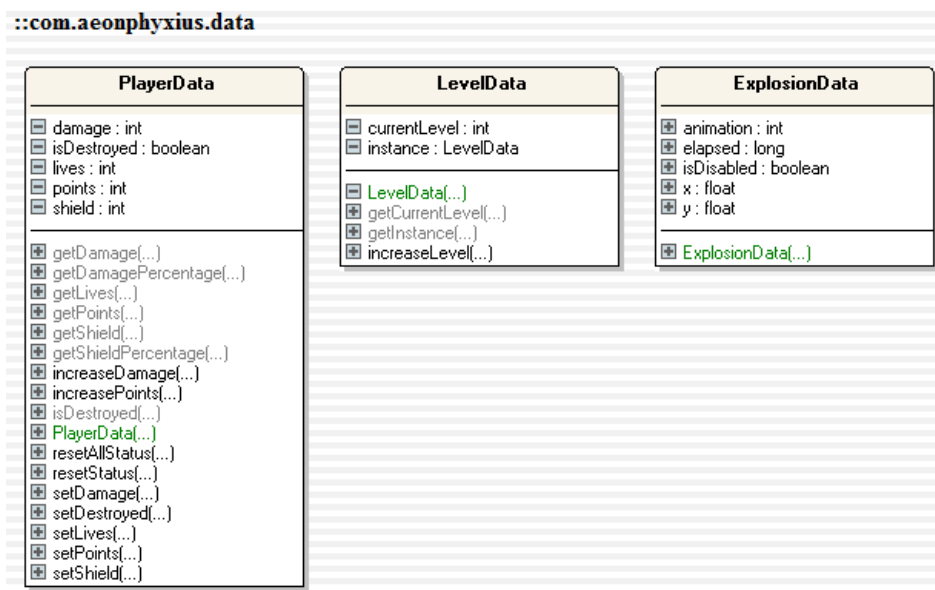


Diagrama de classes 1

Objectes pintats per pantalla

El conjunt de classes que contenen les dades dels objectes que es pintaran per pantalla relacionats amb el joc (Fons, enemics, explosions, jugador i armes).

- Contindran les dades necessàries per tal de pintar a la posició correcta de la pantalla el objecte.
- Estenen de la classe EngineGL, disposaran dels mètodes per manejar tot lo relacionat amb la textura del objecte.
- Algunes d'elles contindran una animació (Explosion) o diferents textures depenen del estat o tipus d'objecte ("Player, Enemy")

Com es pot veure en la imatge a continuació, totes deriven de la classe on disposem dels mètodes per manejar la textura, i pintar-la per pantalla. La majoria son classes contingudes dintre d'una llista per les classes dintre del grup "Manager", exceptuant la "Player", ja que només existeix 1, i "Explosion" que implementa el patró "singleton":

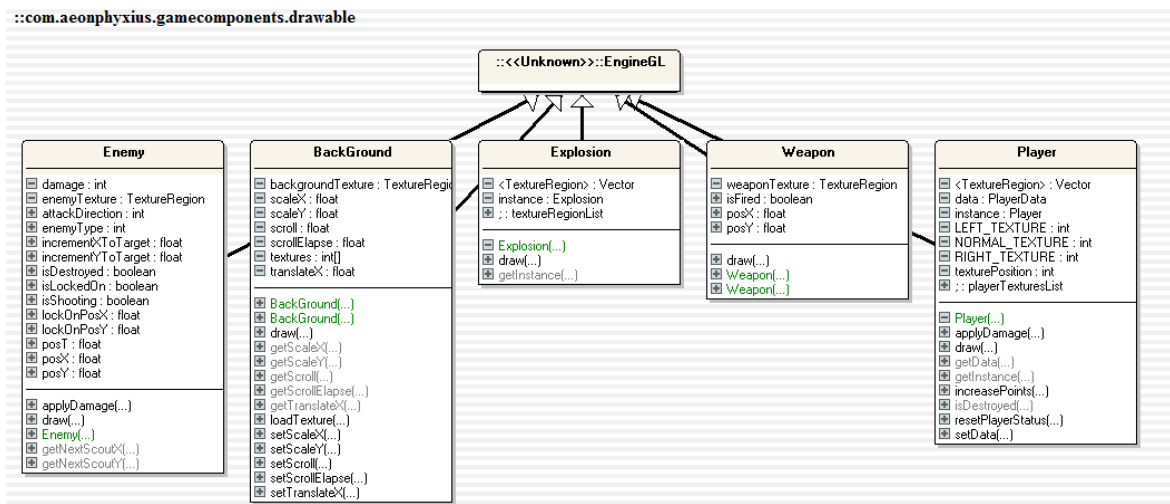


Diagrama de classes 2

Notar també que les dades estan contingudes en aquest nivell (relacionades amb el jugador, els estats dels enemics etc. Per tant, a part de pintar aquests objectes per pantalla (encarregar-se de cridar les funcions adients) també faran una gestió de les dades.

Manegadors de conjunt d'elements (Manager)

Hi ha un conjunt de classes que s'encarreguen de manejar un conjunt d'elements per tal de fer l'actualització a cada cicle. Els elements que controlaran seran els següents:

- Tots els components de la interfície d'usuari del joc (per mostrar al usuari, els punts, el estat dels escuts, etc).
- Manegar els diferents fons del vídeo joc
- Manegar els esquadrans d'enemics, un conjunt d'enemics dintre del esquadró, i carregar les dades del fitxer xml.
- Les armes disparades. Totes les armes disparades tant per part del jugador com dels enemics.
- Totes les explosions que es generaran dintre del vídeo joc.

Com es pot veure en el diagrama següent, la majoria de classes son classes simples, i només en el cas de la carrega del fitxer XML necessitarem estendre la nostra classe. També s'ha de tenir en compte que es fan servir moltes de les classes mencionades en el punt anterior.

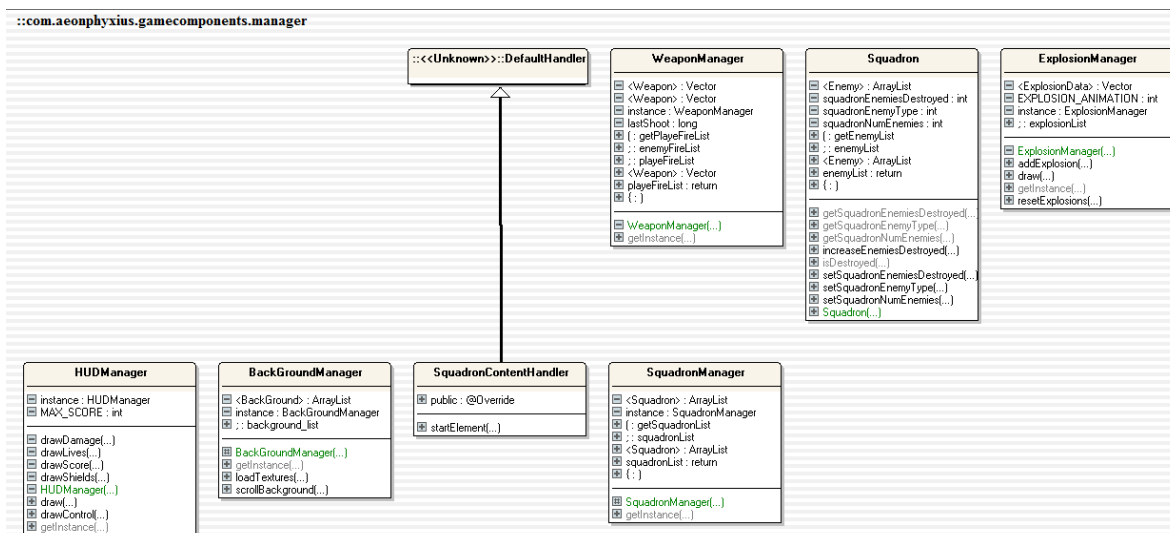


Diagrama de classes 3

Components de la interfície gràfica d'usuari del vídeo joc

La interfície d'usuari del joc consisteix en diferents elements, que mostraran al usuari el estat de la seva nau al vídeo joc, els punts acumulats, la vida, etc. Els diferents components de la interfície gràfica d'usuari seran:

- Imatge amb el numero de vides del jugador (dependent del numero d'aquestes).
- Imatge amb el estat dels escuts.
- Imatge amb el estat de la nau espacial del jugador.
- Punts del jugador.
- Controls de la nau (per que el usuari pugui girar dreta o esquerra)

Com es pot veure en la figura a continuació totes les classes hereten de la classe EngineGL ja que la faran servir per pintar per pantalla els components, i emmagatzemar les dades de la textura de cada objecte:

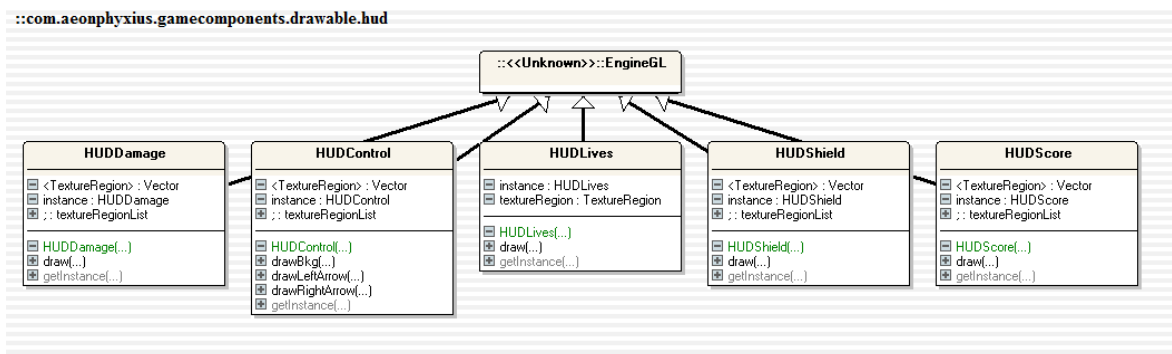


Diagrama de classes 4

Un altre punt important, és que tots implementaran el patró "singleton", ja que només existirà una única instància de cadascun d'ells en el nostre codi, degut, a que només es necessita accedir a ells per pintar una única interfície d'usuari, i per tant es vol per un cantó evitar que es puguin duplicar, i per el altre que l'altre facilitar el seu accés des de qualsevol part del codi.

Pantalles de transició dintre del joc

Dintre del joc hi hauran algunes pantalles de transició entre els diferents estats del vídeo joc. Aquestes seran les següents:

- Des de el menú principal i abans de començar a jugar. Aquesta es repetirà cada cop que el jugador sigui destruït.
- Una entre el moment de la destrucció i el moment de tornar a jugar el nivell.
- Quan es completa un nivell del videojoc
- Quan es termina el videojoc.

Com es pot veure en la imatge a continuació, tots son objectes que es pintaran per pantalla i per tant hereten de la classe EngineGL (per exemple pintar el compte enrere abans de començar a jugar). Per altre cantó per tal de homogeneïtzar el contingut d'aquestes classes s'ha creat la interfície Overlay:

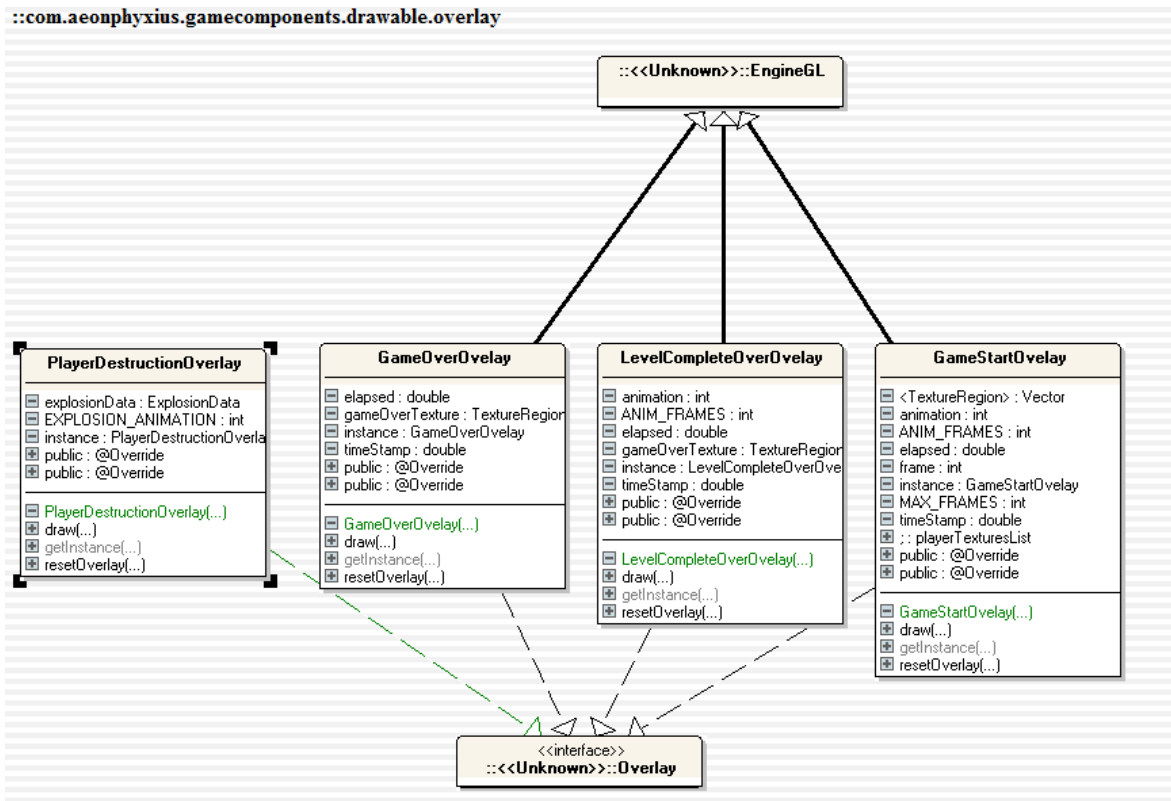


Diagrama de classes 5

El so

El so del vídeo joc estarà encastat dintre de les classes del nucli, en un “package” anomenat “com.aeonphyxius.engine.sound”. Com es pot observar en el diagrama, el so estarà dividit en tres parts:

- Els sons o efectes sonors que es produiran durant el videojoc.
- La musica del videojoc, que serà la que s’escolti de fons.
- La resta de components d’àudio del videojoc.

Totes les classes implementaran una o varies interfícies per tal de clarificar una mica el seu desenvolupament. Com es pot veure a continuació:

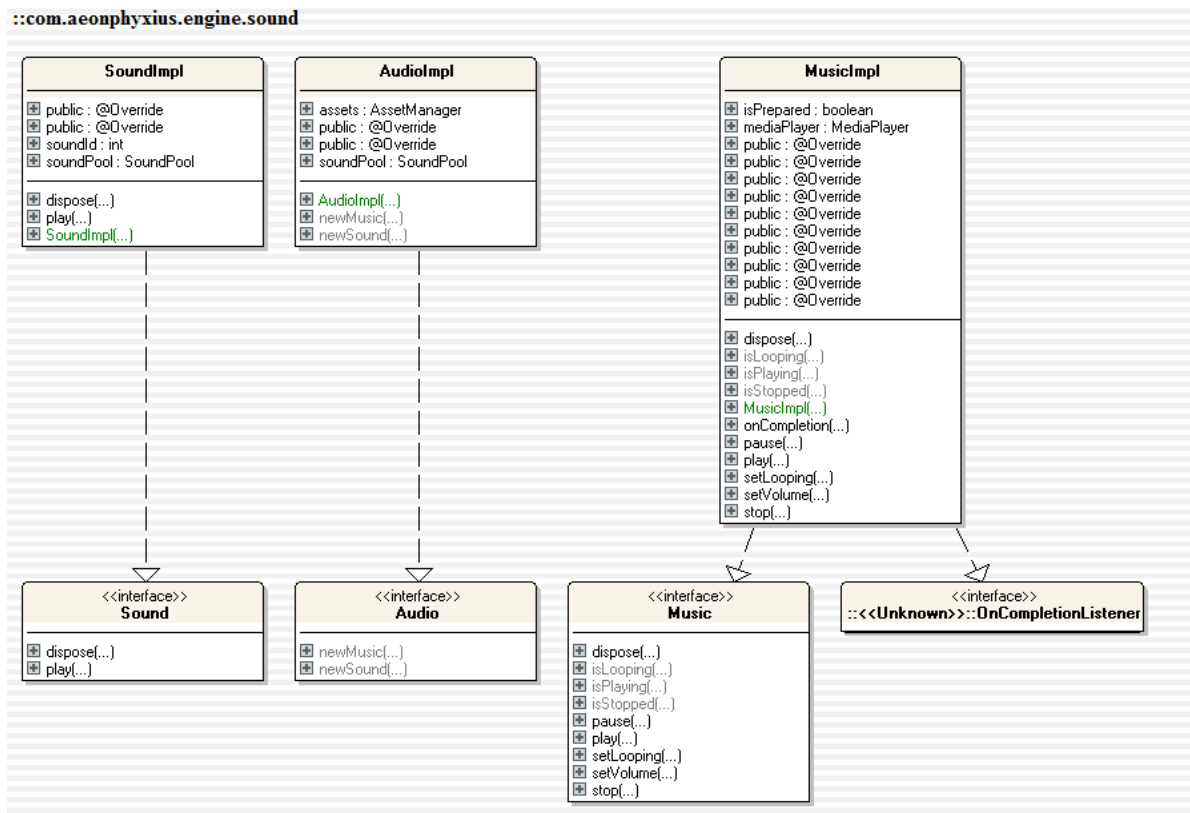


Diagrama de classes 6

Nucli de l'aplicació

El nucli de la nostra aplicació en aquest cas el videojoc estarà concentrat al “package” anomenat “com.aeonphyxius.engine”. Dintre d'aquest podrem trobar manegadors per diferents subsistemes del nostre videojoc:

- Tot lo relacionat amb operacions OpenGL .
- El manegador de la música i la vibració.
- Les variables generals del videojoc
- Utilitats com poden ser la “Bounding Box” o els “Overlay”
- El manegador que s'encarregarà de renderitzar tot el programa, i de la lògica del mateix (“RenderManager”).

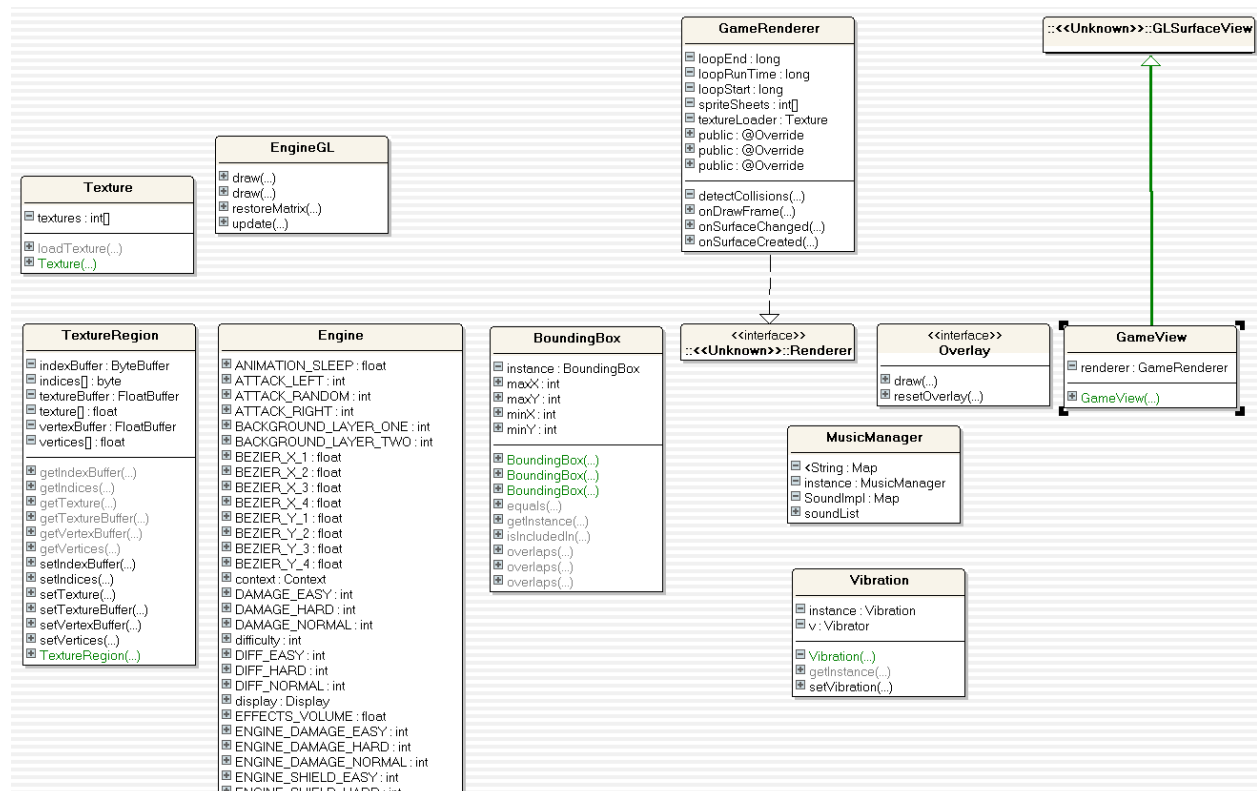


Diagrama de classes 7

Pantalles o “Activity” del videojoc

Com es pot veure en la imatge a continuació el vídeo joc consta de diferents pantalles per mostrar al usuari informació sobre el joc, i es podrien agrupar en 4 tipus:

- Transició: Les pantalles de transició al iniciar el videojoc i al sortir del mateix.
- Menú d’opcions: Els diferents menús d’opcions per ajustar el comportament del videojoc.
- Ajuda: Pantalles d’ajuda o sobre el creador del videojoc.
- El Joc. Aquesta es la pantalla que contindrà el videojoc en si.

Com es pot veure a continuació totes estenen de la classe “android.app.Activity”, i algunes implementen el esdeveniment “onClic” per tal de manegar els esdeveniments rebuts quan el usuari fa clic en els diferents botons:

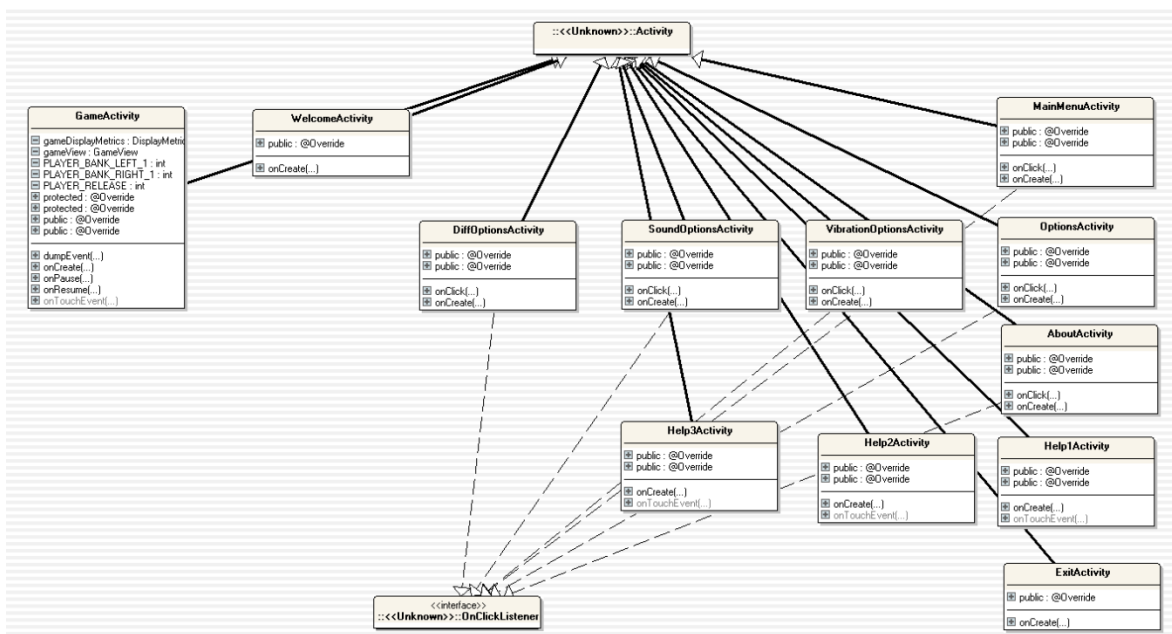


Diagrama de classes 8

Totes les classes fan servir les eines que proporciona Android per tal de definir la disposició i composició de la pantalla, menys la que controla el joc, que crearà una interfície “android.opengl.GLSurfaceView”

El desenvolupament

En aquest apartat cobrirem tot lo relacionat amb el desenvolupament del programari, en aquest cas el videojoc. Es tractaran diferents temes, com la configuració del entorn de programació, el procés de desenvolupament del videojoc, el joc de proves, etc.

Instal·lació i configuració del entorn de programació

En aquest apartat es descriurà breument d'instal·lació i configuració del entorn de programació.

Requisits previs

El únic requisit previ per prepara el entorn de desenvolupament, és disposar d'un ordinador amb Windows XP o superior. Per la realització d'aquest projecte s'han fet servir tant un ordinador de sobre taula i un portàtil, els dos amb "Windows 7 professional Edition".

Instal·lar Java

Per instal·lar la versió de desenvolupament de Java, el primer pas es descarregar-se el instal·lable per Windows de la web oficial

<http://www.oracle.com/technetwork/java/javase/downloads/jdk-7u3-download-1501626.html>

Un cop hem descarregat el instal·lador adient per la nostra versió de sistema operatiu, en el nostre cas la versió serà la JDK 7u3 per Windows x64 (64-bit). Seguim les passes del instal·lador, i ja tindriem instal·lada la versió de desenvolupament de Java.

Instal·lar Eclipse

Per instal·lar Eclipse, un cop instal·lat el entorn de desenvolupament de Java o JDK, serà força senzill. El primer pas serà descarregar-se de la plana web oficial l'última versió del programari. Per aquest projecte hem fet servir la versió Eclipse 3.7.2. La versió que farem servir serà la "Eclipse Classic" ja que serà la que millor s'ajustarà a les nostres necessitats (Hi ha diferents versions, per desenvolupadors J2EE, C++ etc).

Així doncs, ens baixem la versió que necessitem segons la configuració del nostre ordinador, Windows 64 bits en el nostre cas. Ens podem baixar el programari a la següent adreça:

http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.7.2-201202080800/eclipse-SDK-3.7.2-win32-x86_64.zip

Un cop descarregat, descomprimem el directori "eclipse" que hi ha dintre en la localització que desitgem dintre del nostre disc dur. En el nostre cas ho farem a "C:\Program Files"

Nota: No fer servir la localització anterior, si no es disposa de privilegis d'administrador en el usuari que fem servir normalment, per tal d'evitar problemes després quan fem servir l'extensió per Eclipse (problemes de compatibilitat).

Instal·lar ADT (Android Development Tools) i l'extensió per Eclipse

Per instal·lar les ADT, el primer pas es accedir a la web oficial, i descarregar-se l'última versió (<http://developer.android.com/sdk/index.html>), en el nostre cas, hem fet servir la "ADT r18", fent servir el instal·lador per Windows. La instal·lació per Windows es molt senzilla, només caldrà baixar el instal·lador de la pagina que hem vist anteriorment, i seguir les instruccions.

Per més informació sobre la instal·lació, es pot visitar: <http://developer.android.com/sdk/index.html>

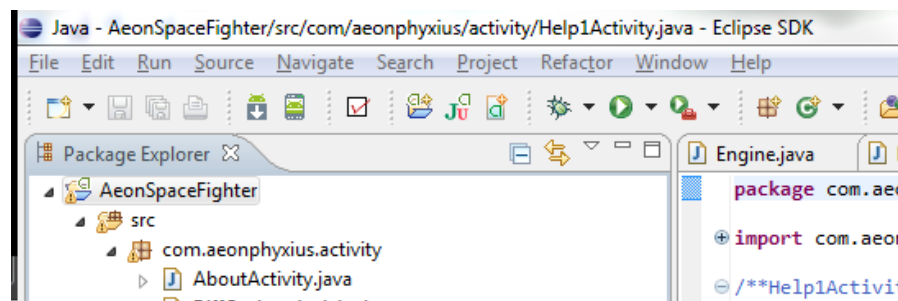
Per últim, un cop ja hem instal·lat el framework de desenvolupament, haurem d'instal·lar l'extensió per Eclipse, per tal de portar a terme el desenvolupament. Hi ha altres entorns de desenvolupament, com utilitzar Netbeans, però oficialment, només està suportat la configuració amb Eclipse. Per instal·lar-ho, les passes seran les següents:

- Obrir el Eclipse i accedir a "Help > Install New Software".
- Afegim un nou rebost, amb la URL <https://dl-ssl.google.com/android/eclipse/>
- En la llista de programari disponible, seleccionarem les "Developer Tools".
- Un cop la instal·lació ha finalitzat, reiniciarem Eclipse.

Potser que després de l'instal·lació, hi hagi que indicar-li a l'extensió on es troba la localització en el disc dur les ADT (que hem instal·lat en el apartat anterior).

Per més informació, es pot visitar: <http://developer.android.com/sdk/eclipse-adt.html#installing>

Així doncs obrint el Eclipse hauríem d'observar les següents icones a la part superior esquerra, com es pot observar a continuació:



Instal·lant 1

Implementació

El següent capítol descriurà el resultat de la implementació de cadascun dels components del videojoc, així com les seves principals característiques, particularitats, problemes trobats durant el desenvolupament.

Les diferents pantalles del videojoc

Com ja s'ha descrit amb anterioritat en aquest document, el videojoc constarà d'un conjunt de pantalles per tal de mostrar al usuari les opcions del videojoc, la pantalla d'entrada i sortida del programa, l'ajuda del mateix, i informació sobre el autor. Hi ha una sèrie de decisions que es van prendre de manera global per totes elles, que enumerarem a continuació:

- El videojoc serà a pantalla completa. Es a dir ocuparà tota la superfície de la pantalla en el nostre dispositiu mòbil o telèfon intel·ligent. Per dur a terme aquesta tasca, en el moment de la creació de les diferents pantalles (Activity) s'ha d'incloure el següent codi:

```
requestWindowFeature(Window.FEATURE_NO_TITLE);  
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

Bàsicament amb aquestes dues línies li estem indicant Android que la pantalla que estem creant serà de pantalla completa (FLAG_FULLSCREEN) i que no voldrem la barra del títol.

- En comptes de desenvolupar un gestor d'esdeveniments per cadascun dels diferents elements de la pantalla (botons), les classes que ho necessitin, implementaran la interfície OnClickListener de la següent manera:

implements OnClickListener

i dintre de la funció detectarem quin component ha generat el esdeveniment de clic i farem la gestió adient, com per exemple dintre de les opcions de so:

```
@Override  
public void onClick(View view) {  
    switch (view.getId()){  
        case R.id.btnBack: // Click on back  
            VibrationManager.getInstance().setVibration(Engine.MENU_CLICK_VIB);  
            SoundOptionsActivity.this.finish();  
            break;  
        case R.id.btnNo: // Click on No sound  
            VibrationManager.getInstance().setVibration(Engine.MENU_CLICK_VIB);  
            Engine.isMuted = true;  
            // Closes this activity  
            SoundOptionsActivity.this.finish();  
            break;  
        case R.id.btnYes: // Click on YES sound  
            VibrationManager.getInstance().setVibration(Engine.MENU_CLICK_VIB);  
            Engine.isMuted= false; // Set sound to true  
            // Closes this activity  
            SoundOptionsActivity.this.finish();  
            break;  
    }  
}
```

- Totes les pantalles (exceptuant la de jugar) tindran la seva disposició definida en el fitxer de "layout". Aquesta contindrà tota la definició de la mateixa, la imatge de fons, la col·locació dels botons, etc. Totes estaran contingudes al directori res/layout del projecte, i normalment el seu nom serà explicatiu de a quina pantalla pertany. Per fer aquest disseny Android disposa d'una interfície gràfica, però donat els problemes que dona el seu disseny amb aquesta (es definiran a la secció de problemes), es va preferir fer amb el codi XML directament. El editor de "layouts" amb la pantalla d'opcions de so, es veuria de la següent manera:



Desenvolupament 1

Com es pot veure en la imatge anterior, es pot fer el disseny de manera gràfica. Un altre curiositat, es que el efecte dels botons quan es fa clic sobre ells s'ha programat amb el codi de cadascuna de les pantalles, així com d'altres opcions, com per exemple en el codi de la mateixa pantalla esmenada anteriorment:

```

// Set menu button options
ImageButton back = (ImageButton)findViewById(R.id.btnBack);
ImageButton yes = (ImageButton)findViewById(R.id.btnYes);
ImageButton no = (ImageButton)findViewById(R.id.btnNo);

// Set alpha and haptic feedback for all the button's image
back.setBackground().setAlpha(Engine.MENU_BUTTON_ALPHA);
back.setHapticFeedbackEnabled(Engine.HAPTIC_BUTTON_FEEDBACK);

yes.setBackground().setAlpha(Engine.MENU_BUTTON_ALPHA);
yes.setHapticFeedbackEnabled(Engine.HAPTIC_BUTTON_FEEDBACK);

no.setBackground().setAlpha(Engine.MENU_BUTTON_ALPHA);
no.setHapticFeedbackEnabled(Engine.HAPTIC_BUTTON_FEEDBACK);

// Add onClick listeners to this menu buttons
yes.setOnClickListener(this);
no.setOnClickListener(this);
back.setOnClickListener(this);
  
```

- La pantalla del videojoc en comptes de definir el seu disseny igual que tota la resta, ho farà implementant el "android.opengl.GLSurfaceView" que manejarà tota la implementació de la part OpenGL del videojoc.
- Totes les classes que pertanyen a aquest grup, estaran al "package" package com.aeonphyxius.activity, ja que dintre de la nomenclatura d'Android, Activity son les pantalles.
- Tots els esdeveniments "onClick" dels diferents botons a totes les pantalles del videojoc, si el so i la vibració estan activats, reproduirà un petit so de "clic" i una petita vibració el dispositiu.

A continuació mostrarem les pantalles del videojoc, així com una petita descripció de les mateixes i les funcionalitats que ofereixen cadascuna d'elles:

- Benvinguda (Welcome): Aquesta pantalla donarà la benvinguda als jugadors, estarà uns segons, i automàticament, passarà a la següent pantalla, de pantalla principal del videojoc. Aquest procés es farà utilitzant el rellotge intern del dispositiu, per tant hauria de tenir la mateixa durada en tot tipus de dispositius (velocitats de processament) Aquesta pantalla s'utilitza normalment en els videojocs, per mostrar el logotip del estudi / s de programació que ho han desenvolupat. En aquest cas, hi ha el meu sobrenom, i el meu logotip:



Desenvolupament 2

- Menú principal (MainMenu): Aquesta serà la primera pantalla amb la qual podrem interactuar un cop hem executat el videojoc. Només constarà de 5 esdeveniments "onClick" per les cinc diferents opcions, que estudiarem a continuació (menys la de jugar que s'estudia en propers capítols):



Desenvolupament 3

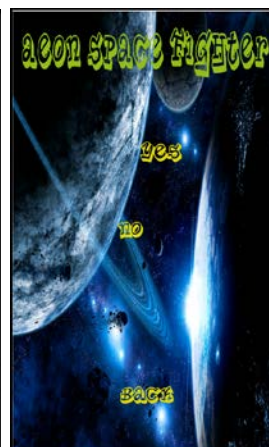
- Opcions (Options): Quan es fa clic en el boto opció en el apartat anterior, passarem a una pantalla on ens mostraran els tres elements que podem configurar en el nostre videojoc, so / vibració / dificultat (fig: desenvolupament 4). Un cop triat quina opció volem configurar, tornarem sempre a aquesta pantalla, i si no es vol configurar cap opció sempre es disposarà d'un boto per tornar enrere o "back" (fig: desenvolupament 5,6 i 7):



Desenvolupament 4



Desenvolupament 5

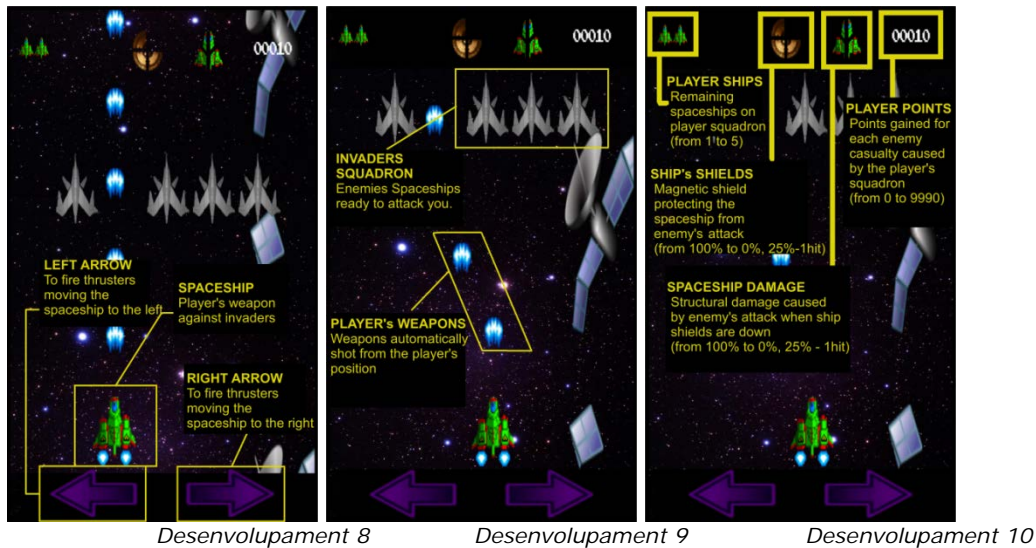


Desenvolupament 6



Desenvolupament 7

- Ajuda (Help): En cas de fer clic en el menú principal, a la opció d'ajuda, el usuari accedirà a una successió de tres pantalles on es mostrarà informació de com jugar al nostre videojoc. La transició es farà mitjançant un esdeveniment onClick a qualsevol punt de la pantalla. Un cop vist la tercera pantalla d'ajuda, el usuari retornarà al menú principal. Les pantalles son les que podem veure a continuació:



La major diferencia que s'ha implementat en les "Activity" de les pantalles d'ajuda, es que en comptes de tenir en compte quin component ha generat el esdeveniment de la següent manera:

```

@Override
public boolean onTouchEvent(MotionEvent event) {
    Help1Activity.this.finish();
    return true;
}

```

- Sortida (Exit): Aquesta pantalla donarà el adéu als jugadors, estarà uns segons, i automàticament, sortirà del videojoc, com es pot veure en la següent imatge:



Desenvolupament 11

- Jugar (Play): La pantalla del joc, la descriurem en els punts a continuació

El jugador

El jugador, i tot lo relacionat amb aquest (puntuació, nau espacial, estatus, ...) és una de les parts principals del desenvolupament d'un videojoc. El correcte funcionament d'aquesta part serà bàsica per una bona experiència d'usuari, i per tant de la valoració que faran aquests del nostre videojoc.

Els enemics

Per implementar tota la lògica dels enemics, s'han fet servir principalment tres classes; una per manegar tots els esquadrans d'enemics, un altre per manegar cada esquadró, i l'ultima per manegar tota la informació dels enemics.

- **SquadronManager:** Al començament de cada nivell, aquesta classe carregarà tota la llista d'esquadrans des de un fitxer XML, on el nom dependrà del nivell (level1.xml, level2.xml....). Aquest fitxer serà de la següent manera:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<squadrans>
  <squadron ypos="6" enemy="1" dir="0" xpos1="0" xpos2="2" xpos3="4" xpos4="4" xpos5="5"/>
  <squadron ypos="7" enemy="1" dir="0" xpos1="0" xpos2="3" xpos3="4" />
  <squadron ypos="8" enemy="1" dir="0" xpos1="0" xpos2="3" xpos3="6" />
  <squadron ypos="9" enemy="1" dir="0" xpos1="0" xpos2="3" xpos3="4" />
  <squadron ypos="10" enemy="1" dir="0" xpos1="0" xpos2="1" xpos3="2" xpos4="3"/>

  <squadron ypos="12" enemy="2" dir="0" xpos1="1" xpos2="3" xpos3="4" />
  <squadron ypos="13" enemy="2" dir="0" xpos1="1" xpos2="3" xpos3="4" />
  <squadron ypos="14" enemy="2" dir="0" xpos1="1" xpos2="3" xpos3="4" />
  <squadron ypos="15" enemy="2" dir="0" xpos1="1" xpos2="3" xpos3="4"/>
  <squadron ypos="16" enemy="2" dir="0" xpos1="1" xpos2="3" xpos3="4"/>
  <squadron ypos="17" enemy="2" dir="0" xpos1="1" xpos2="3" xpos3="4"/>
  <squadron ypos="18" enemy="2" dir="0" xpos1="1" xpos2="3" xpos3="4"/>

  <squadron ypos="20" enemy="1" dir="0" xpos1="0" xpos2="3" xpos3="4" />
  <squadron ypos="22" enemy="1" dir="0" xpos1="1" xpos2="3" xpos3="4"/>
  <squadron ypos="24" enemy="3" dir="0" xpos1="0" xpos2="3" xpos3="4" />
  <squadron ypos="26" enemy="2" dir="0" xpos1="0" xpos2="3" xpos3="4" xpos4="5" />
  <squadron ypos="28" enemy="1" dir="0" xpos1="1" xpos2="3" xpos3="4"/>
  <squadron ypos="30" enemy="1" dir="0" xpos1="0" xpos2="3" xpos3="4" />

  <squadron ypos="32" enemy="3" dir="0" xpos1="0" xpos2="2" xpos3="4"/>
  <squadron ypos="33" enemy="3" dir="0" xpos1="0" xpos2="2" xpos3="4"/>
  <squadron ypos="34" enemy="3" dir="0" xpos1="1" xpos2="3" xpos3="5"/>

```

Com es pot veure, cada línia serà un nou esquadró on l'indiquem el numero d'enemics i el seu tipus. Tota la lògica per fer l'actualització dels esquadrans esfera dintre d'aquesta classe:

- **Squadron:** Aquesta classe es molt senzilla i només contindrà la lògica per manegar una llista d'enemics.
- **Enemy:** Aquesta classe conté tota la lògica del enemic, de manera individual, els seus escuts, i tots els estats en general del mateix

La interfície d'usuari

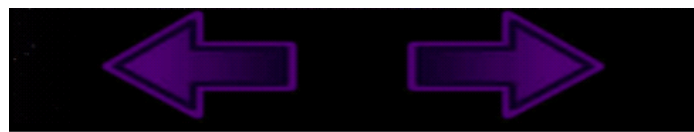
La interfície d'usuari es la part del nostre videojoc que s'encarregarà de mostrar al jugador el estat general del joc. Per portar a terme aquesta tasca, es van desenvolupar un objecte per cada component de la interfície d'usuari, i un de general que els conte a tots, i s'encarrega d'instanciar-los i administrar-los.

Tots els components de la interfície gràfica d'usuari es trobaran dintre del "package" "package com.aeonphyxius.gamecomponents.drawable.hud;" i seran els següents:

- **HUDControl:** Aquest component s'encarregarà de tota la gestió de mostrar al usuari el control. Es a dir, s'encarregarà de mostrar al usuari, la part inferior de la pantalla, on es mostren dues fletxes a mode de control, i s'il·lumina la que esta pitjada per el usuari en cada moment. Per dur a terme aquesta tasca, estendran de la classe EngineGL per poder fer servir les utilitats de pintat per pantalla, i implementarà el patró de disseny singleton, ja que nomes voldrem una instancia d'aquesta classe en el nostre projecte, de la manera:

```
public class HUDControl extends EngineGL{  
  
    private static HUDControl instance = null;           // Singleton implementation  
    private Vector<TextureRegion> textureRegionList;    // Texture region containing the icon to show  
  
    /**
```

I el resultat de la renderització d'aquest element serà:

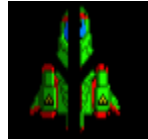


HUD 1

- **HUDDamage:** Aquest component s'encarregarà de mostrar de manera gràfica el percentatge de dany estructural de la nau espacial. El component serà una icona de la nau, dividida en 4 parts (25% cadascuna). Per dur a terme aquesta tasca, estendran de la classe EngineGL per poder fer servir les utilitats de pintat per pantalla, i implementarà el patró de disseny singleton, ja que nomes voldrem una instancia d'aquesta classe en el nostre projecte, de la manera:

```
public class HUDDamage extends EngineGL{  
  
    private static HUDDamage instance = null;           // Singleton implementation  
    private Vector<TextureRegion> textureRegionList;    // Texture region containing the icon to show
```

i el resultat de la renderització d'aquesta classe serà:

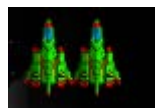


HUD 2

- **HUDLives:** Per mostrar al usuari el número de vides que encara té disponible, hem decidit fer servir icones amb la figura de la nau, una figura per vida disponible. Per dur a terme aquesta tasca, estendran de la classe EngineGL per poder fer servir les utilitats de pintat per pantalla, i implementarà el patró de disseny singleton, ja que només voldrem una instància d'aquesta classe en el nostre projecte, de la manera:

```
public class HUDLives extends EngineGL {  
    private static HUDLives instance = null; // Singleton implementations  
    private TextureRegion textureRegion; // Texture region containing the icon to show  
  
    /**  
     * Singleton implementation of the unique instance of this class  
     *  
     * @return  
     */  
    public static HUDLives getInstance() {  
        if (instance == null) {  
            instance = new HUDLives();  
        }  
        return instance;  
    }  
}
```

i el resultat de la renderització d'aquesta classe serà:



HUD 3

- **HUDScore:** Per mostrar aquesta tasca, s'han renderitzat els números una a un, ja que no es disposa d'una manera millor per fer-ho amb OpenGL. Després a cada cicle s'han de separar les xifres del número per tal de representar la xifra adient per pantalla. Per dur a terme aquesta tasca, estendran de la classe EngineGL per poder fer servir les utilitats de pintat per pantalla, i implementarà el patró de disseny singleton, ja que només voldrem una instància d'aquesta classe en el nostre projecte, de la manera:

```

public class HUDScore extends EngineGL {
    private static HUDScore instance = null; // Singleton implementations
    private Vector<TextureRegion> textureRegionList; // Texture region containing the icon to show

    /**
     * Singleton implementation of the unique instance of this class
     * @return
     */
    public static HUDScore getInstance() {
        if (instance == null) {
            instance = new HUDScore();
        }
        return instance;
    }
}

```

I per el càlcul de quina xifra s'ha de col·locar en cada posició:

```

int maxValue = 10000;
int countStringPos = 0;
String tempScore = Integer.toString(Player.getInstance().getData().getPoints());

for (int i=0;i<MAX_SCORE;i++) { // loop the 5 numbers that compose the scores

    gl.glMatrixMode(GL10.GL_MODELVIEW);
    gl.glLoadIdentity();
    gl.glPushMatrix(); // Save Matrix before transformations
    gl.glScalef(.025f, .025f, 1f); // Scale the original image
    gl.glTranslatef(+32.0f, 37.5f, 0f); // Position on screen (-0.1) because at 0 there is a small space at the bottom
    gl.glMatrixMode(GL10.GL_TEXTURE); // Texture Mode
    gl.glLoadIdentity();

    // Select the correct texture (number) to display
    if (Player.getInstance().getData().getPoints()< maxValue){
        HUDScore.getInstance().draw(gl, spriteSheet,0);
    }else{
        HUDScore.getInstance().draw(gl, spriteSheet,Integer.parseInt(tempScore.charAt(countStringPos)+"") );
        countStringPos++;
    }

    maxValue= maxValue/10; // Reduce to next digit

    gl.glPopMatrix(); // Recover previous Matrix
    gl.glLoadIdentity();
}

```

i el resultat de la renderització d'aquesta classe serà:



- HUDShield:** Aquest component s'encarregarà de mostrar de manera gràfica el percentatge de dany en els escuts de la nau espacial. El component serà una icona d'un escut, dividit en 4 parts (25% cadascuna). Per dur a terme aquesta tasca, estendran de la classe EngineGL per poder fer servir les utilitats de pintat per pantalla, i implementarà el patró de disseny singleton, ja que només voldrem una instància d'aquesta classe en el nostre projecte, de la manera:

```

public class HUDShield extends EngineGL{
    private static HUDShield instance = null; // Singleton implementation
    private Vector<TextureRegion> textureRegionList; // Texture region containing the icon to show

    /**
     * Singleton implementation of the unique instance of this class
     * @return
     */
    public static HUDShield getInstance() {
        if (instance == null) {
            instance = new HUDShield();
        }
        return instance;
    }
}

```

i el resultat de la renderització d'aquesta classe serà:



HUD 5

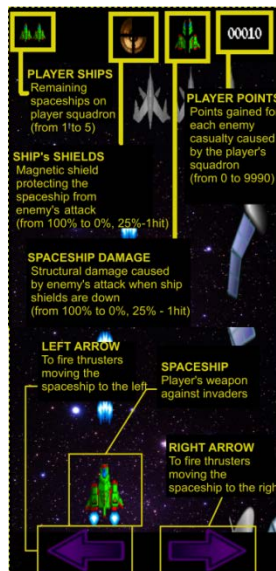
Per últim el component que s'encarregarà de manejar tots, i que s'actualitzin de manera correcta a cada cicle de pantalla, serà el HUDManager. Hi haurà un mètode que serà el que es cridi des de el RenderManager, per tal d'actualitzar cadascun dels elements, serà:

```

/**
 * HUD draw method to display all components on screen
 * @param gl OpenGL handler
 * @param spriteSheet array containing all sprites ids
 */
public void draw(GL10 gl, int[] spriteSheet) {

    drawControl (gl, spriteSheet);
    drawLives (gl, spriteSheet);
    drawShields(gl, spriteSheet);
    drawDamage (gl, spriteSheet);
    drawScore (gl, spriteSheet);
}
  
```

El resultat final doncs seria el següent:



HUD 6

El so

En la part del so, es podrien diferenciar dues parts importants, la musica i els sons que es reproduïxen depenent del esdeveniment. A continuació descriurem breument la implementació de les dues parts:

- La musica:** Per reproduir un arxiu de música hi ha diverses opcions en Android, però en aquest cas, he decidit fer servir el reproductor multimèdia del sistema o "MediaPlayer". Es pot carregar un fitxer d'àudio directament en la classe MediaPlayer de la següent manera:

```

public void play(AssetFileDescriptor assetDescriptor) {
    mediaPlayer = new MediaPlayer();
    try {
        mediaPlayer.setDataSource(assetDescriptor.getFileDescriptor(),
            assetDescriptor.getStartOffset(),
            assetDescriptor.getLength());
        mediaPlayer.prepare();
        isPrepared = true;
        mediaPlayer.setOnCompletionListener(this);
    } catch (Exception e) {
        throw new RuntimeException("Couldn't load music");
    }
}

```

A continuació s'han d'implentar tots els mètodes que vam definir a la interfície, per tal d'obligar la inclusió dels mètodes que podem necessitar (Play, Stop, Resume, Pause, ...). El format acceptat per Android en els fitxers d'àudio és ogg, que és un format lliure i que pertany a la organització <http://xiph.org/>

- El so:** Per el desenvolupament de la part de reproducció dels sons, he decidit crear un fitxer xml amb tots els fitxers que carregarem, i el seu sobrenom dintre del codi, per tal que es pugui cridar des de qualsevol punt del nostre projecte, amb una unica linea. El fitxer XML seria d'aquesta manera:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<sounds>
  <sound name="coin" src="coin.ogg"/>
  <sound name="blaster" src="blaster.ogg"/>
  <sound name="blaster" src="blaster.ogg"/>
  <sound name="finalEnemy" src="finalEnemy.ogg"/>
  <sound name="explosion" src="explosion.ogg"/>
  <sound name="explosionEnemy" src="explosionEnemy.ogg"/>
  <sound name="fushionShot" src="fushionshot.ogg"/>
  <sound name="laserHit" src="laser_hit.ogg"/>
  <sound name="click" src="click.ogg"/>
  <sound name="clickBack" src="clickBack.ogg"/>
</sounds>

```

I es carregarà fent servir un XML parser dintre del nostre codi. Un cop ja el tenim a memòria (tots els arxius son en format ogg), ja estarà disponible per que a qualsevol altre part del codi, s'executi algun so. La manera serà gracies a la implementació el patró de disseny singleton de la manera:

```
*/  
  
public class MusicManager {  
  
    private static MusicManager instance = null;  
    private Map<String, SoundImpl> soundList;  
    private MusicImpl music;  
  
    /**  
     * Singleton pattern implementation  
     * @return the unique instance of the Music Manager  
     */  
    public static MusicManager getInstance() {  
        if (instance == null) {  
            instance = new MusicManager();  
        }  
        return instance;  
    }  
}
```

Gracies a aquesta implementació des de qualsevol punt cridarem un so, de la manera:

```
Vibration.getInstance().setVibration(Engine.MENU_CLICK_VIB);  
MusicManager.getInstance().playSound(Engine.SOUND_CLICK);
```

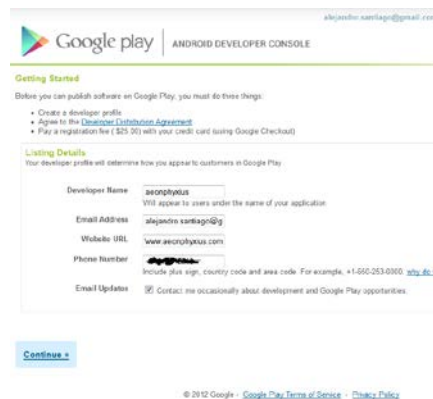
Publicant el projecte en el "Google Play"

Un cop ja s'ha terminat el desenvolupament i les proves del videojoc, el últim pas serà distribuir el producte obtingut entre els possibles jugadors. Hi ha dues opcions de les quals, en aquest projecte només farem servir la segona:

- Distribuir el fitxer APK resultant de la compilació del projecte.
- Publicar el nostre videojoc en la "Google Play".

Registrar-se com a desenvolupador oficial

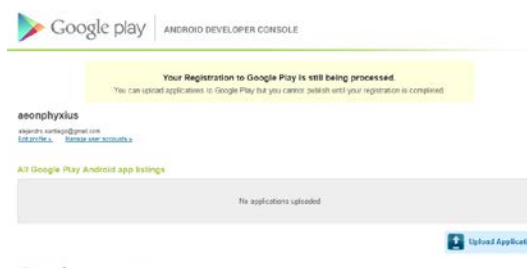
El primer pas per poder publicar aplicacions al Google Play, es registrar-se com a usuari en aquest servei, i pagar una contribució de 25\$ (dòlars americans). El procés es força senzill, i en un parell de clics de ratolí, i utilitzant el meu compte de correu electrònic de Google, i el meu compte de Google Checkout. Com es pot veure en la captura de pantalla següent, no és un procés gaire complicat:



The screenshot shows the 'Getting Started' section of the Google Play Android Developer Console. It lists three steps: 'Create a developer profile', 'Agree to the Developer Distribution Agreement', and 'Pay a registration fee (\$25 US) with your credit card using Google Checkout'. Below this is the 'Listing Details' form with fields for Developer Name (aeonphyxius), Email Address (alejandre.santiago@y), Website URL (www.aeonphyxius.com), Phone Number, and Email Updates (checked).

Publicant 1

Un cop realitzat el pagament, rebrem una confirmació en el compte de correu que hem indicat, el en nostre compte a Google Play ja estarà creada, però pendent de confirmació i activació per part de Google com es pot veure en la imatge següent(en el meu cas va trigar 6 hores més o menys en ser aprovada):



Publicant 2

Preparant el binari per la seva distribució

Un cop ja soc membre el Google Play com a desenvolupador oficial de la plataforma, el següent pas es preparar el resultat del projecte per poder-lo publicar. El primer pas es signar l'aplicació i per dur aquest procés s'han de fer les següents comprovacions prèvies:

- Remoure el atribut "*android:debuggable*" del tag "*<application>*" en el fitxer de "*manifest*" de la nostra aplicació.

- Assignar un valor als següents atributs dintre del nostre arxiu de *"manifest"*:
 - *"android:versionCode"*: Si ja hem publicat una versió anterior de la nostra aplicació haurem d'incrementar aquest valor. En cas contrari, donar-li un valor inicial.
 - *"android:versionName"*: Igual que el atribut anterior, donar-li un nom diferent a cadascuna de les versions del nostre projecte.
 - *"android:installLocation"*: Si voldrem instal·lar en el emmagatzemament extern si està disponible, alliberant així el espai intern del telèfon.
 - *"android:targetSdkVersion"* i *"android:minSdkVersion"*: Amb aquest dos atributs s'han de definir els SDK al qual va dirigit la nostra aplicació així com la versió mínima. Així en el "Google Play" la nostra aplicació només apareixerà als telèfons que continguin compleixin aquesta condició i per tant podran fer servir la nostra aplicació.
- Assegurar-nos de donar a l'aplicació només els permisos que realment necessitem, i no posar d'extres que no es necessiten per el correcte funcionament d'ela mateixa. Als usuaris no els hi agrada que una aplicació demani permisos no necessaris. Per dur a terme aquesta tasca, s'han d'examinar les etiquetes *"<uses-permission>"*

Signant l'APK resultant del projecte

Un cop ja s'ha preparat el codi per a la seva distribució, ja estarem en disposició de preparar el binari que pujarem a la web del "Google Play" per a la seva distribució, lo que es coneix com a signar el APK del nostre projecte. Les passes per aconseguir aquest objectiu seran les següents:

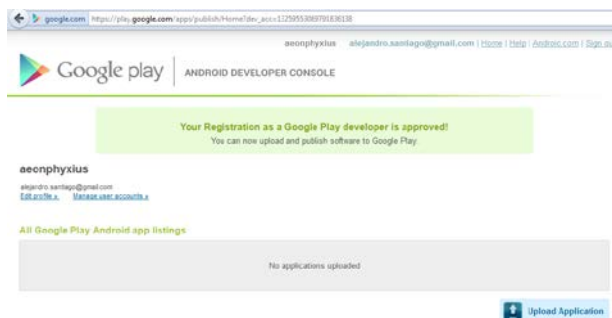
- Seleccionarem el directori on volem crear el fitxer APK.
- Haurem d'usar una de les claus que ja hem creat prèviament en el nostre ordinador (o fer-ho durant aquest procés).

Un cop seguit aquestes dues passes, ja tindrem la nostra aplicació preparada per publicar-la en Google Play. Per començar aquest procés s'ha de seleccionar "Export signed application package" en el menú contextual de "Android tools" en el nostre projecte.

Publicant Aeon Space Fighter en "Google Play"

Un cop ja hem preparat la nostra aplicació (videojoc en aquest cas) per la seva distribució, el últim pas que ens queda per realitzar es pujar la nostra aplicació en "Google Play". Com es podrà veure a continuació les passes son molt senzilles. Així doncs a continuació detallarem quin son aquestes passes:

- El primer pas serà accedir a la nostra pagina de desenvolupador a “Google Play” en l’adreça (<https://play.google.com/apps/publish/>). En aquesta adreça accedirem a lo que es coneix com la consola del desenvolupador.

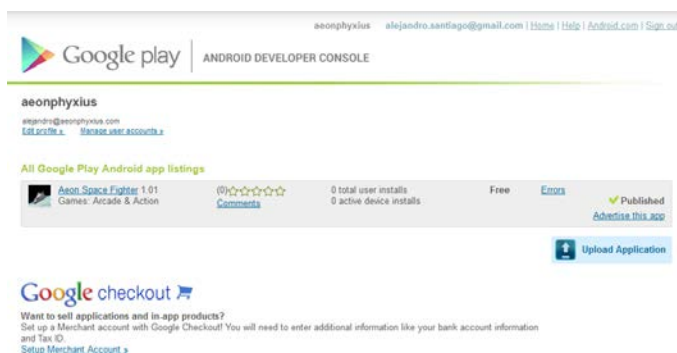


Publicant 3

- Un cop en la nostra consola, hi ha una opció a sota a la dreta per tal de pujar una aplicació. A part de seleccionar en el nostre disc dur local el fitxer apk amb l’aplicació, haurem també d’afegir la següent informació obligatòriament:
 - Captures de pantalla (“Screenshoots”): Com a mínim s’hauran de pujar dos captures de pantalla, en el format i mida especificats.
 - Icona d’alta resolució: Una imatge de la nostra aplicació de 512x512 a mode d’icona.
 - El llenguatge de l’aplicació: si escau es poden configurar més llenguatges a part del que hi ha per defecte (angles en el nostre cas).
 - Títol: El títol o nom de la nostra aplicació.
 - Descripció: Una petita descripció del nostre projecte per tal de que els usuaris sàpiguen que
 - Canvis recents: En cas que hi hagin noves actualitzacions utilitzarem aquesta informació per explicar-los.
 - Tipus d’aplicació: si es un videojoc o una aplicació
 - Categoria: dintre del tipus d’aplicació trobarem una llista de subtipus o categories.
 - Protecció de copia: Triar si volem que es pugui copiar entre diferents dispositius o no.
 - Contingut: Tipus de contingut del programa o videojoc (segons edats, etc).

- Preu: Si volem cobrar per el producte, o gratis com serà el nostre cas.
- Paisos: Hi ha una llista de països que podem triar on publicar la nostra aplicació. En el nostre cas, la publicarem en tota la llista.
- Informació de contacte: Contacte sobre el autor de l'aplicació, en aquest cas les meves dades de contacte.

Un cop publicada l'aplicació podrem veure com ja esta publicada en la consola de desenvolupador, i en el llistat d'aplicacions podrem veure la que hem creat:



Publicant 4

Com a anècdota, comentar que el videojoc tardarà al voltant d'unes 2 hores en esta disponible al Google Play, així que aquest procés no es instantani.

Seguiment de l'aplicació

Un cop hem publicat el nostre videojoc en Google Play, i els usuaris comencin a descarregar la mateixa, començaran a aparèixer els errors. Com en tot procés de desenvolupament de programari, un cop s'entrega l'aplicació als usuaris finals, l'aparició d'errors serà comú i per tant haurem de fer un seguiment dels mateixos.

Google Play disposa d'una eina per tal de que els usuaris que vulguin participar, quan l'aplicació dona algun error, enviar un informe o "crash report". En cas que decideixin contribuir, enviaran un informe al creador de l'aplicació, i nosaltres el rebrem com es pot veure a la següent imatge:

alejandrosantiago@aeonphyxius.com
[Edit profile](#) [Manage user accounts](#)

All Google Play Android app listings

 Aeon Space Fighter 1.03 Games: Arcade & Action	(0)☆☆☆☆☆ Comments	0 total user installs 0 active device installs	Free	Errors (1)	✔ Published Advertise this app
--	--------------------------------------	---	------	----------------------------	---

 Upload Application

Seguiment 1

Si entrem a veure els detalls del error, veurem la traça que ha deixat aquest:

java.lang.ArrayIndexOutOfBoundsException in com.aeonphyxius

Crash

Exception class	java.lang.ArrayIndexOutOfBoundsException
Source method	Vector.arrayIndexOutOfBoundsException()

[Show user messages](#)

Mark as old

Platforms

OTHER	2 reports	2 reports/week
-------	-----------	----------------

Stack traces

v1.01	Jun 8, 2012 12:16:36 AM	2 reports	2 reports/week
-------	-------------------------	-----------	----------------

```
java.lang.ArrayIndexOutOfBoundsException: length=3; index=3
at java.util.Vector.arrayIndexOutOfBoundsException(Vector.java:907)
at java.util.Vector.elementAt(Vector.java:328)
at java.util.Vector.get(Vector.java:442)
at com.aeonphyxius.gamecomponents.drawable.overlay.GameStartOvelay.draw(GameStartOvelay.java:84)
at com.aeonphyxius.engine.GameRenderere.onDrawFrame(GameRenderere.java:64)
at android.opengl.GLSurfaceView$GLThread.guardedRun(GLSurfaceView.java:1462)
at android.opengl.GLSurfaceView$GLThread.run(GLSurfaceView.java:1216)
```

Seguiment 2

En aquest cas, el seguiment d'aquest error m'ha permès arreglar un error que hi havia al canviar de nivell, gracies a gent que s'ha baixat el videojoc i ha estat jugant. Un cop solucionat el error, he publicat una nova versió arreglant el mateix.

Resum

Com he pogut constatar, publicar una aplicació d'Android per el públic en general es força senzill. Per un canto, pagar la llicencia son 25 dòlars i s'ha de fer una vegada només. Aquest punt crec que encara que no és car (uns 20 dòlars al canvi actual), farà que només usuaris realment interessats en publicar el seu treball pagaran aquest preu. Com he vist en el procés de publicació del meu projecte, un cop pagat i enregistrat com a desenvolupador, pujar la nostra aplicació es qüestió d'uns pocs clics de ratolí.

Conclusions

En aquest última apartat de la memòria, exposaré les meves conclusions personals, un cop el projecte està enllestit, la presentació i el vídeo amb ella estan finalitzats, i estic escrivint aquest últim capítol de la memòria. Per tant, a escasses hores de fer l'entrega final, i que pot ser la meva última entrega de la Enginyeria Superior en Informàtica, estic en disposició de fer una valoració global d'aquest projecte.

Abans de començar el projecte les meves expectatives eren més ben baixes, ja que no sabia si podria triar el projecte que volia dur a terme, i en cas afirmatiu, no sabia el podria fer de l'abast que tenia pensat. Veia que seria molt complicat dur a terme un videojoc complet en només 3 mesos, amb tot el aprenentatge sobre al nova plataforma, i la nova tecnologia, encara ho complicava més. Però abans de començar els ànims sempre son alts, i els repte era bonic.

Durant el desenvolupament del mateix ha hagut moments de voler llençar la tovallola, per veure que era impossible complir amb els termes establerts i entregar un videojoc funcionant a principis de Juny. Entre la tercer entrega i la quarta vaig estudiar seriosament abandonar, i continuar en el següent semestre, però finalment vaig decidir al menys intentar-ho, i veure si podria arribar al final. Durant el procés de desenvolupament van sorgir molts problemes a nivell tècnic per el meu desconeixement de la nova plataforma i això va fer endarrerir considerablement el projecte.

El procés d'enginyeria del programari, ja sabia a priori que és un procés molt complicat, i que moltes vegades és molt difícil portar a terme els projectes seguint la planificació establerta, i havent-hi estudiat totes les possibles variables. En el desenvolupament del meu PFC, m'he trobat amb molts problemes per aconseguir arribar a les dates d'entrega, tant per motius personals com laborals. Lo que sobre el paper semblava molt senzill d'aconseguir, a la realitat ha estat molt més complicat del que havia planificat. El desenvolupament, a més a més ha estat molt més llarg, i amb algunes complicacions amb las que no en contava al inici del projecte, i no podia ni entreveure que apareixerien. Si jo que només era un equip d'una persona, he tingut aquests problemes, he de reconèixer que quan parlem d'equips molt més grans, crec que ha de ser una tasca gaire impossible el complir totes les fites del projecte en les dates acordades, per la gran quantitat de factors que poden influir en el mateix.

Sobre el desenvolupament en la plataforma Android, he trobat llums i ombres en tot lo relacionat amb la mateixa. Per un cantó, m'he trobat que el "framework" de desenvolupament es molt complert, hi ha molta ajuda a la xarxa, així com exemples de cadascuna de les funcionalitats a la mateixa pàgina de desenvolupament de Android (cosa que sempre he trobat a faltar a la API de JAVA). També crec que els dos llibres sobre aquesta temàtica que vaig adquirir han estat un gran ajut ja que tots els conceptes i el codi font adjunt han estat de força ajut. Però també m'he trobat alguna ombra en tot aquest procés, com per exemple el emulador del que disposa les ADT es força dolent, i

personalment crec que no està a la altura de la resta de components, complicant de vegades el desenvolupament i la depuració dels projectes. Altre punt fosc, és la obligació de fer servir el Eclipse com a IDE de desenvolupament oficial. Encara que es poden fer servir d'altres, el nivell d'integració més alt només es podrà aconseguir amb aquest. Així doncs, crec que el "framework" de desenvolupament es força complet, però espero que en les properes versions millorin substancialment aquests petits inconvenients, ja que si el comparem amb d'altres de la competència, surt clarament perdedor d'aquesta comparació.

Des d'un punt de vista més subjectiu, el desenvolupament d'aquest projecte ha estat molt enriquidor a nivell personal per una banda i tècnic per l'altre. Com ja he comentat al llarg d'aquesta memòria, un dels motius pel qual volia portar a terme aquest projecte, era el aprofundir tècnicament en el desenvolupament de videojocs per una plataforma mòbil. Crec que durant els pròxims anys, el oci digital en aquestes plataformes experimentarà una gran expansió i potser m'agradaria poder dirigir el meu futur professional en aquesta direcció. El altre gran motiu per portar a terme aquest projecte, i que per tant ha estat molt enriquidor al llarg del mateix, es la meva afició (passió més ben dit), per la programació de videojocs. Fa molts anys que mica en mica vaig encaminant el meu futur professional cap aquest sector, i aquest projecte de ben segur, m'ha enriquit en aquest aspecte. Així doncs com a resum del mateix, crec que he adquirit nous coneixements sobre la planificació de projectes i els problemes associats a l'execució dels mateixos per un cantó, i tècnicament he aprofundit en els meus coneixements sobre Android i les plataformes mòbils, així com el desenvolupament de videojocs.

Finalment, i ja escrivint les meves ultimes línies d'aquesta memòria, crec que al final he aconseguit disposar d'un videojoc complert en el termini establert, un producte del que hem sentit orgullosos, i que ha valgut la pena el esforç realitzat.

Bibliografia

En aquest capítol veurem una llista dels llibres que he utilitzat per la realització d'aquest projecte, amb les seves dades (editorial, ISBN, títol, autor,...) i un petit comentari de la informació que buscava a cada llibre, que vaig trobar, i si he fet servir part del codi com referència.

Els llibres

Títol	Practical Android 4 Games Development		
Autor	J.F DiMarzio	ISBN	978-1-4302-4030-3
Editorial	Apress		
Informació	Desenvolupament de videojocs Android		
Títol	Beginning OpenGL game programming		
Autor	Dave Astle / Kevin Hawkins	ISBN	978-1-59200-369-3
Editorial	Thompson		
Informació	Programació d'OpenGL aplicada al món dels videojocs		
Títol	Programación en OpenGL (Una guía de referencia completa de OpenGL)		
Autor	Richard S.Wright Jr./Michael Sweet	ISBN	84-415-0176-9
Editorial	Anaya multimèdia		
Informació	Programació d'OpenGL i guia de referència		
Títol	Beginning Android 4 Games Development		
Autor	Mario Zechner / Robert Green	ISBN	978-1-4302-3987-1
Editorial	Apress		
Informació	Desenvolupament de videojocs Android		

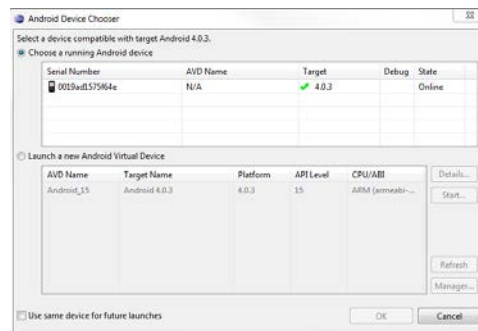
Annex

Com fer servir el videojoc

Samsung Galaxy S II

Per tal de poder fer servir el videojoc en un dispositiu amb Android (en aquest cas el Samsung Galaxy S II) Hi tindrem dues opcions diferents:

- Per un canto com hem vist en capítols anteriors de la memòria, podrem accedir a ells a través de la botiga oficial al Google Play, busquem el joc per el nom “Aeon Space Fighter” el descarreguem i instal·lem, i ja podríem jugar.
- L'altre opció es executar el codi font d'aquesta practica. Per fer això, hem d'importar el projecte amb Eclipse, i un cop configurat com hem vist anteriorment internarem executar el projecte, amb lo que ens demanarà a on o volem instal·lar.



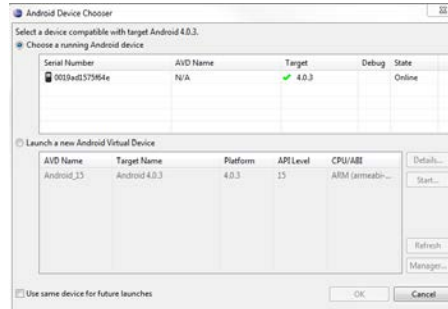
Samsung Galaxy S II 1

Seleccionarem el nostre dispositiu, que com es pot veure en la imatge anterior, serà el únic que esta connectat al nostre ordinador. Un cop el seleccionem, i tenim activat la opció de “usb debugging” al nostre dispositiu mòbil, fem clic a “OK” i instal·larà automàticament el videojoc al dispositiu mòbil.

Emulador

Per tal de fer servir el videojoc amb el emulador que proporciona les ADT, el procés serà molt semblant a fer-ho servir amb un dispositiu real, des de el Eclipse (no hi ha accés al Google Play des de el emulador).

El primer pas serà executar el projecte des de el Eclipse, i en la finestra que s'obra per triar el dispositiu en la que volem executar el videojoc, triem la opció de “Launch a new Android Virtual Device” i acceptem aquesta finestra.



Emulador 1

Un cop això, començarà la carrega del “Virtual Device”, que pot arribar a trigar 5 min aproximadament dependent de la velocitat del nostre ordinador. Un cop carregat, el projecte que estem executant, començarà a executar-se de manera automàtica, amb lo que anirem directament al menú principal del videojoc:



Emulador 2

Si per exemple provem de jugar al videojoc, veurem que l’execució del mateix en el “Virtual Device” es extremadament lent, per lo que no ens servirà com a plataforma de proves. Aquest es un problema conegut amb el emulador, però el qual no te solució, així que com a proves ens serveix, però no per gaire més (veure vídeo de la presentació).



Emulador 3