



**Universitat Autònoma
de Barcelona**

TriviaRace

Joc XNA per a Xbox 360

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Sistemes

realitzat per

Miquel Sitges Blasco

i dirigit per

Xavier Verge Mestre

Escola d'Enginyeria

Sabadell, *juny* de 2012

El sotasignat, **Xavier Verge Mestre**,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció per

Miquel Sitges Blasco

I per a que consti firma la present.
Sabadell, **juny** de **2012**

Signat: **Xavier Verge Mestre**

Títol del projecte: TriviaRace: Videojoc XNA per a Xbox 360	
Autor: Miquel Sitges Blasco	Data: <i>Juny 2012</i>
Tutor: Xavier Verge Mestre	
Titulació: Enginyeria Tècnica en Informàtica de Sistemes	
Paraules clau <ul style="list-style-type: none">• <u>Català:</u> XNA, Xbox 360, videojoc, videoconsola, TriviaRace, jugador, 3D, avatars, IA, carrera, pregunta, resposta, qüestionari.• <u>Castellà:</u> XNA, Xbox 360, videojuego, videoconsola, TriviaRace, jugador, 3D, avatares, IA, carrera, pregunta, respuesta, cuestionario.• <u>Anglès:</u> XNA, Xbox 360, videogame, video console, TriviaRace, gamer, 3D, avatars, AI, race, question, answer, questionnaire.	
Resum del projecte <ul style="list-style-type: none">• <u>Català:</u> Aquest projecte documenta la realització d'un videojoc anomenat TriviaRace per a la consola Xbox 360.<p>Els jugadors han de competir per ser els primers en arribar al final de l'escenari i contestar correctament a una pregunta que se'ls formula. Per arribar-hi abans que els seus contrincants, poden utilitzar objectes per a molestar-los. Poden jugar 4 jugadors simultàniament, ja siguin controlats per persones reals o per la consola, mitjançant una senzilla intel·ligència artificial.</p><p>El desenvolupament del joc s'ha realitzat mitjançant XNA, unes eines de Microsoft orientades a la creació de videojocs per a diverses plataformes, inclosa la consola Xbox 360.</p>• <u>Castellà:</u> Este proyecto documenta la realización de un videojuego llamado TriviaRace para la consola Xbox 360.<p>Los jugadores deben competir para ser los primeros en llegar al final del escenario y contestar correctamente a una pregunta que se les formula. Para llegar antes que sus contrincantes, pueden utilizar objetos para molestarlos. Pueden jugar 4 jugadores simultáneamente, ya sean controlados por personas reales o por la consola, mediante una sencilla inteligencia artificial.</p><p>El desarrollo del juego se ha realizado mediante XNA, unas herramientas de Microsoft orientadas a la creación de videojuegos para varias plataformas, incluida la consola Xbox 360.</p>• <u>Anglès:</u> This project documents the realization of a videogame called TriviaRace for the Xbox 360.<p>Players must compete to be the first to reach the end of the stage and correctly answer the question they are made. To arrive before the other players, they can use objects to tease them. The game can be played by 4 players at the same time, controlled by real people or the console, using a simple artificial intelligence.</p><p>Game development was performed with XNA, some Microsoft videogame-oriented tools for various platforms, including the Xbox 360.</p>	

Índex

1. Introducció	5
1.1 Descripció del projecte	5
1.2 Guió del joc	6
1.3 Estat de l'art	8
1.4 Estructura de la memòria	9
2. Estudi de viabilitat	11
2.1 Introducció	11
2.1.1 Tipologia i paraules clau	11
2.1.2 Objectius del projecte	11
2.1.3 Priorització dels objectius	12
2.1.4 Definicions, acrònims i abreviacions	12
2.2 Parts interessades	15
2.2.1 Stakeholders	15
2.2.2 Perfils d'usuaris	15
2.2.3 Equip del projecte	15
2.3 Requisits del sistema	15
2.3.1 Requisits funcionals (RF)	15
2.3.2 Requisits no funcionals (RNF)	18
2.3.3 Priorització dels requisits	18
2.3.4 Relació dels requisits amb els objectius	19
2.4 Alternatives	19
2.4.1 Alternativa 1	20
2.4.2 Alternativa 2	20
2.4.3 Alternativa 3	21
2.4.4 Alternativa 4	21
2.4.5 Alternativa 5	22
2.4.6 Solució proposada	22
2.5 Planificació	23
2.5.1 Tasques i fites	24
2.5.2 Diagrama de Gantt	28
2.6 Avaluació dels riscos	30
2.6.1 Identificació de riscos	30
2.6.2 Catalogació dels riscos	30
2.6.3 Pla de contingència	31
2.7 Pressupost	31
2.7.1 Costos de personal	31
2.7.2 Costos dels recursos	31

2.7.3	Costos d'amortització	31
2.8	Conclusions de l'estudi de viabilitat	32
3.	Fonaments teòrics	33
3.1	Xbox 360	33
3.2	Microsoft XNA	35
3.3	XACT	37
3.4	C#	38
3.5	Visual Studio	38
4.	Anàlisis i disseny del joc	41
4.1	Diagrama dels casos d'ús	41
4.2	Especificació dels casos d'ús	41
4.3	Inici i desenvolupament de la idea	49
4.4	Pantalles de títol, opcions i tenda.....	51
4.5	Pantalla de joc	65
5.	Proves i errors.....	77
5.1	Proves	77
5.2	Erroros durant el desenvolupament	78
6.	Conclusions	81
6.1	Compliment d'objectius	81
6.2	Desviacions	81
6.3	Línies d'ampliació	82
6.4	Conclusions finals	83
6.5	Agraïments.....	83
7.	Bibliografia.....	85
8.	Annexos	87
8.1	Annex I: Manual d'usuari.....	87
8.2	Annex II: Blog del projecte	88

Índex de figures

Figura 1: Xbox 360 Elite (esquerra), Xbox 360 S (dreta)	33
Figura 2: Diagrama de dependències entre classes.....	51
Figura 3: Classe TriviaRaceGame.....	52
Figura 4: Classe Player.....	53
Figura 5: Avatars	54
Figura 6: Classe IA	54
Figura 7: Classe PlayerInput	55
Figura 8: Classe Items	56
Figura 9: Classes AvatarCustomAnimation i CustomAvatarAnimationData.....	57
Figura 10: Animació personalitzada de desmai	58
Figura 11: Classe TitleScreen	58
Figura 12: Pantalla TitleScreen.....	59
Figura 13: Pantalla MenuScreen	59
Figura 14: Classe MenuScreen	60
Figura 15: Classe Menu.....	62
Figura 16: Classe ShopScreen.....	63
Figura 17: Pantalla ShopScreen.....	65
Figura 18: Classe GameScreen.....	66
Figura 19: Pantalla GameScreen preparada per a tres jugadors	67
Figura 20: Barra de la interfície gràfica	68
Figura 21: Pantalla GameScreen amb els noms de cada jugador	70
Figura 22: Classe Preguntes.....	70
Figura 23: Classe Scene	72
Figura 24: Classe SceneObject	73
Figura 25: Model 3D de font en el programa Blender	74
Figura 26: Zona de respostes de GameScreen	75
Figura 27: Pantalla d'error de gestió en les càmeres multijugador	78

1. Introducció

Una de les idees que sempre he volgut dur a terme ha estat el desenvolupament d'un videojoc, i amb el projecte de final de carrera he vist l'oportunitat de realitzar-la. Aquesta idea em permetria mostrar la maduresa d'un projecte viable per a la seva comercialització, així com mostrar els coneixements apresos durant aquests anys de carrera. A més, desenvolupar un videojoc permet mantenir una motivació constant, ja que és un procés molt creatiu i amè de portar a terme. L'elecció del videojoc s'ajusta plenament dins de la definició de projecte de final de carrera ja que:

- Per a desenvolupar un videojoc és necessari treballar en apartats diferents de forma que tots encaixin perfectament: mecànica de joc, disseny, so, etc. Un cop acabat, el projecte podria ser la base per a un desenvolupament major, a vistes de poder ser comercialitzat.
- La programació ha jugat un paper molt important en la carrera i en el videojoc, així com també els gràfics 3D i la gestió de les diverses fases del projecte.
- Els videojocs tenen actualment una posició molt important en la indústria de l'oci, i és interessant veure com s'estructura internament un videojoc i tota la feina que porta darrere. A més, és una feina molt creativa que permet anar tant lluny com es vulgui, el que garanteix una motivació per anar superant-se mica en mica durant el desenvolupament del joc.

A nivell personal, un dels objectius que he volgut dur a terme ha estat aprendre a utilitzar a fons les eines XNA en les que es basa el projecte i poder provar-lo en una videoconsola comercial actual. Així, a més de portar endavant el projecte de final de carrera, també puc adquirir nous coneixements que poden resultar-me útils en un futur.

1.1 Descripció del projecte

Quan algú pensa en un videojoc, el més normal és que la primera cosa que li passi pel cap sigui la imatge d'alguna superproducció moderna, o d'algun videojoc que hagi estat anunciat per televisió, o fins i tot d'aquell joc antic que va jugar fa tant de temps. Tots aquests videojocs tenen un enfocament molt comercial i la majoria disposen d'un equip de moltes persones darrere, així com d'un gran pressupost, que permeten uns resultats espectaculars. Tot i això, no tots els videojocs són així.

A part dels grans videojocs comercials, també existeixen els anomenats videojocs independents, o *indie*. Aquests jocs tenen com a característiques principals que no són finançats per una distribuïdora de videojocs i que l'equip que els desenvolupa és molt reduït, d'una a deu persones com a molt.

Aquesta limitació de recursos fa que moltes vegades els resultats obtinguts no siguin tan espectaculars com els que poden obtenir les grans produccions, però el fet de no estar lligat a una productora comporta que no hi hagi interessos comercials imposats o limitacions creatives en el seu desenvolupament. Per això, els desenvolupaments indie acostumen a destacar per les seves aportacions innovadores (ja sigui en la manera de jugar o en l'apartat artístic i creatiu), molt més arriscades que els jocs més comercials, que necessiten assegurar les vendes.

Per al joc desenvolupat en aquest projecte s'ha buscat la semblança en l'estètica i el comportament dels jocs indie més que els comercials, degut a les limitacions de recursos. A més, també s'ha de tenir en compte el temps d'aprenentatge en la utilització de les eines i el funcionament del sistema. Degut a això també sorgeix la limitació temporal.

Tot això comporta que l'objectiu final del projecte sigui aconseguir una base similar a la d'un videojoc indie, sobre la que es pugui treballar un cop acabat el projecte, però que per si sola tingui el suficient contingut i demostrï els coneixements apresos durant la carrera.

Primer de tot es va fer una pluja d'idees per tal de decidir quin tipus de videojoc era el més adequat per a desenvolupar. Un cop decidit el tipus i muntat un guió de joc, es va decidir donar èmfasi a la part de programació més que a la resta d'apartats ja que ha tingut molta més rellevància dins la carrera i per això és preferible que se li dediqui més temps. Per aquest motiu s'intenta donar al joc en si el màxim de característiques possibles i un apartat més auster en els camps de disseny d'escenaris i de so.

1.2 Guió del joc

L'objectiu del joc és que els jugadors arribin a la meta que es troba al final de l'escenari i siguin els primers en respondre correctament la pregunta que es formula, mentre competeixen contra els altres jugadors, controlats per altres persones o pel mateix joc, de forma simultània.

Cada jugador comença la partida amb uns punts determinats d'atac i de defensa, amb els que, a la tenda d'objectes, pot comprar ítems per atacar a altres jugadors o defensar-se dels seus atacs, i d'aquesta manera tenir més facilitats per arribar abans a la meta.

A l'iniciar el joc tots els jugadors que volen jugar han d'iniciar sessió a la consola. Si a la partida s'afegeixen menys de quatre persones, els personatges que queden fins a arribar a quatre seran controlats per la intel·ligència artificial del joc. Una vegada iniciades les sessions, s'escull la categoria de preguntes que es faran durant la partida i el nivell que tindran els personatges controlats per la màquina. Un cop escollida la categoria i el nivell de la intel·ligència artificial, cada jugador pot comprar a la tenda d'objectes els ítems que consideri més oportuns per preparar la primera partida. Quan tothom ha comprat el que ha volgut amb els punts dels que disposa, pot començar el joc.

Al començar el joc els quatre jugadors apareixen al principi de l'escenari preparats per sortir corrents. A la vegada es mostra la pregunta que hauran de contestar a l'arribar al final de l'escenari. A partir d'aquest moment els jugadors ja poden moure's per l'escenari. El primer jugador que arribi a la resposta correcta guanyarà més punts d'atac que el segon, i així consecutivament. Si els jugadors són atacats durant la partida, guanyen punts de defensa. Els punts que es guanyin es poden utilitzar més endavant quan s'acabi la partida i els jugadors puguin tornar a passar per la tenda d'objectes.

Un cop els jugadors arriben a la zona de respostes, cada jugador ha d'escollir una de les tres possibles respostes i entrar dins de la zona de la resposta que consideri correcta. Per a que la resposta sigui vàlida, el jugador ha de quedar-se dins de la zona de resposta cinc segons. Durant aquests cinc segons ha d'impedir que els altres jugadors el facin fora, havent de començar el compte enrere de nou si ho aconsegueixen.

Quan tots els jugadors han escollit una resposta s'acaba la primera ronda i es fa la distribució de punts, segons si els jugadors han encertat la resposta correcta i en quina posició han arribat. Llavors comença una nova ronda del joc amb els jugadors altre cop a l'inici de l'escenari i amb una nova pregunta que contestar. Un cop s'han jugat tres rondes, s'acaba la partida i amb els punts obtinguts durant aquestes es decideix qui guanya.

Cada tres preguntes es torna a la tenda d'ítems per comprar més objectes. Hi ha quatre tipus d'ítems:

- Atac 1: Treu un 50% de l'energia dels jugadors més propers.
- Atac 2: Treu un 40% de l'energia de tots els jugadors.
- Defensa 1: Recupera 50 punts d'energia.
- Defensa 2: Evita els atacs durant 10 segons.

Així doncs, aquest projecte és, a més de joc, una possible eina educativa. Mitjançant les preguntes es pot utilitzar el joc com a prova de coneixements sobre una matèria, com pot ser un tema d'una assignatura d'aquesta carrera. D'aquesta forma, els jugadors es converteixen en alumnes i aprenen conceptes o en confirmen els ja adquirits, camuflats en un videojoc. Madurant la idea bàsica d'aquest projecte i ampliant-la, es pot arribar a un videojoc educatiu viable a nivell comercial. Els alumnes juguen i aprenen alhora.

1.3 Estat de l'art

Actualment, la indústria del videojoc és una de les indústries més rentables, sobrepassant en alguns països fins i tot les de cinema i música juntes. La majoria de videojocs d'èxit, però, provenen de les grans companyies que gràcies als elevats recursos disponibles i la gran maquinària de màrqueting poden crear jocs més complexos i vistosos i donar-los a conèixer a un públic més ampli.

A nivell més reduït, ja sigui a través de petites empreses o de desenvolupadors independents, al no poder disposar de tants recursos econòmics, la majoria de títols que treuen queden més amagats. En aquests casos, el que millor funciona és el boca a boca quan un joc val la pena i agrada al públic.

En el cas d'aquest projecte, un joc de preguntes en el que el jugador ha de competir contra la resta de participants, ser el primer en arribar a la zona de respostes i contestar de forma correcta, el mercat no té cap producte similar conegut, si bé hi ha parts de la mecànica que podem trobar en diversos videojocs.

Quan pensem en videojocs de preguntes, els títols més exitosos avui dia són sense cap dubte les múltiples versions del joc de taula *Trivial Pursuit*, com *Trivial Pursuit Wii* o *Scene It?*, o jocs que emulen un concurs de televisió, com *TV Show King* o *Buzz!*. En aquests casos, el jugador no interactua gaire amb el joc, havent de contestar simplement les preguntes que se li formulen, sense tenir control sobre el seu personatge. Només una de les versions de *Buzz!* titulada *Buzz! HQ* i

llançada per a la xarxa social de la consola PlayStation 3 *Home*, el jugador ha de moure el seu personatge fins a la zona que correspon a la resposta correcta en un temps determinat, disposant d'una mica més de interactivitat amb el joc.

Per altra banda, el fet d'utilitzar ítems per atacar o defensar-se d'altres jugadors és una característica comuna a una gran varietat de títols que poc tenen a veure entre ells. Per exemple, a qualsevol de les vàries versions del joc *Super Mario Kart* els jugadors han d'aconseguir objectes durant una carrera de karts per tenir avantatge respecte la resta de jugadors i arribar abans a la meta. També en el joc *Super Smash Bros.*, on el personatge ha de lluitar contra altres i els jugadors poden aprofitar-se dels poders que atorguen els diferents objectes que hi ha per l'escenari.

Cap d'aquests títols, però, encaixa dins de la descripció del joc que ocupa aquest projecte, que és una mescla de tots aquests elements, i que a més de ser educatiu, dóna un plus de interactivitat amb el jugador que permet que no es cansi del joc gaire aviat.

1.4 Estructura de la memòria

La memòria del projecte està estructurada en varis apartats. El primer de tots és la introducció, on es fa un petit resum del que tractarà el projecte.

El segon apartat és l'estudi de viabilitat. Aquí es fa un repàs a les bases del projecte, es mostra com està definit i el perquè s'ha decidit desenvolupar-lo de la forma en què s'ha fet i perquè no s'ha prioritzat altres alternatives. També s'hi troba un historial de les diverses versions que s'han llançat del joc durant el seu desenvolupament.

A continuació, s'expliquen els fonaments teòrics del projecte, aprofundint en les eines que s'han utilitzat.

El següent apartat explica la mecànica del joc i les decisions de disseny que s'han realitzat. També es mostren els casos d'ús i els diagrames de classes del projecte.

En el proper apartat es fa un seguit de les proves que s'han realitzat per comprovar que el joc no falli i no tingui cap error. També s'analitzen els errors més importants que han sorgit durant el desenvolupament i com s'han solucionat.

Al final de la memòria es troben les conclusions del projecte, així com el compliment d'objectius, la desviació que ha sofert la planificació durant el desenvolupament i les possibles línies d'ampliació que es podrien aplicar al joc.

Per acabar es troba la bibliografia que s'ha utilitzat per al desenvolupament del joc i s'adjunta un annex amb una guia ràpida per poder jugar al joc i el recull d'entrades del diari del projecte.

2. Estudi de viabilitat

2.1 Introducció

L'estudi de viabilitat ha de garantir que el projecte es pugui dur a terme amb èxit, tenint en compte què és el que es vol aconseguir, quins són els riscos i quines les possibles alternatives. Això ha de facilitar la presa de decisions a l'hora de desenvolupar el joc i prendre totes les precaucions possibles per evitar el màxim d'errors o desviacions del projecte possibles.

2.1.1 Tipologia i paraules clau

Tipologia: Projecte orientat al desenvolupament de software.

Paraules clau: XNA, Xbox 360, videojoc, videoconsola, TriviaRace, jugador, 3D, avatars, IA, carrera, pregunta, resposta, qüestionari.

2.1.2 Objectius del projecte

L'objectiu principal del projecte és desenvolupar un videojoc donant més pes a la part de programació que als apartats de disseny gràfic i so. Tenint això en compte, els objectius es poden dividir en:

- O.1. Dissenyar i programar un joc multijugador.
- O.2. Gestionar les sessions de varis jugadors de forma simultània.
- O.3. Afegir una interfície gràfica per a cada jugador.
- O.4. Afegir música de fons i efectes de so.
- O.5. Implementar una intel·ligència artificial simple.
- O.6. Utilitzar els avatars de la Xbox 360 com a personatges.
- O.7. Afegir animacions personalitzades per als avatars.
- O.8. Crear diversos escenaris per a jugar durant les partides.
- O.9. Implementar el qüestionari de preguntes mitjançant un arxiu XML.
- O.10. Crear diversos menús per a personalitzar la partida.
- O.11. Afegir objectes per atacar o defensar-se durant la partida.
- O.12. Crear una tenda on es puguin comprar els objectes.
- O.13. Implementar un sistema de puntuació segons els resultats obtinguts.
- O.14. Implementar un sistema d'energia per a cada jugador.

2.1.3 Priorització dels objectius

A la taula que es presenta a continuació es vol representar la priorització dels objectius anteriors que s'ha tingut en compte a l'hora de planificar el projecte.

	Crític	Prioritari	Secundari
O.1.	X		
O.2.	X		
O.3.		X	
O.4.			X
O.5.	X		
O.6.		X	
O.7.			X
O.8.		X	
O.9.			X
O.10.	X		
O.11.	X		
O.12.		X	
O.13.		X	
O.14.	X		

2.1.4 Definicions, acrònims i abreviacions

Alpha: Una versió alpha és la primera fase d'un software, generalment encara inestable, però que ja té suficients característiques finals implementades com per començar a fer-li proves.

API (Application Programming Interface): Una API és una interfície que permet accedir a les funcions i procediments d'un servei a través d'un software de tercers. Depenent de com estigui preparada la API es podrà accedir a més o menys serveis.

App Hub: App Hub és una comunitat de Microsoft on s'hi troben desenvolupadors per preguntar dubtes sobre el software que estan desenvolupant, enviar les aplicacions i jocs per a ser revisats i poder-los publicar. Per poder publicar una aplicació per a Xbox 360, per exemple, és necessari ser membre d'App Hub, entre altres requisits.

Avatar: Un avatar de Xbox 360 és una representació gràfica de l'usuari que imita la forma humana. Els usuaris poden personalitzar els trets característics del seu

avatar, així com la roba que vesteixen. Alguns jocs permeten utilitzar l'avatar de cada jugador com a personatge.

Bazar: El Bazar Xbox Live és un mercat virtual desenvolupat per Microsoft, que permet als usuaris comprar i descarregar continguts per a la consola, tals com jocs XBLA, indie o comercials, demostracions de jocs i contingut DLC descarregable. També té una secció de música i una altra de pel·lícules digitals. Depenent del contingut que l'usuari vulgui comprar, s'han d'utilitzar diners reals o punts virtuals de Microsoft per fer-ho.

Beta: Una versió beta és la fase en el desenvolupament d'un software on ja estan implementades totes o la gran totalitat de característiques finals. En aquest tipus de versions tan sols falta depurar el codi i buscar errors per a que el software sigui el més estable possible.

Betatester: Un betatester és una persona encarregada de provar un software en fase beta, trobar tots els errors possibles, i fer un informe detallant cada un d'ells. Gràcies a aquests informes, els desenvolupadors del software poden solucionar-los més ràpidament.

Brainstorming: També conegut com a "pluja d'idees", és el procés creatiu que recull les idees d'un grup reduït de persones per a dur a terme un projecte o buscar una solució. Aquestes idees són comentades després pels membres i es busca quina és la millor o millors opcions.

Bug: Terme anglès que s'utilitza per descriure un error en el software.

Gameplay: Terme anglès per descriure la jugabilitat en un videojoc. El gameplay és la forma amb la que els jugadors interactuen amb el joc a través de les seves normes i funcionament, sense tenir en compte altres parts com l'àudio o els gràfics.

Gamertag: És la fitxa personal de cada jugador. En ella s'hi guarda varis tipus d'informació, com el nom del jugador o el seu avatar.

IA (Intel·ligència Artificial): És el conjunt de tècniques informàtiques que fan que un sistema sigui conscient del seu entorn i actuï buscant el resultat més positiu possible.

Joc local: En l'àmbit dels videojocs, el joc local és el que es juga directament amb la consola sense cap tipus de connexió amb l'exterior. Un joc multijugador local requereix que tots els jugadors juguin en el mateix sistema alhora.

Joc online: A diferència del joc local, el joc online és el tipus de videojoc en el que cada jugador juga des d'un sistema diferent però dins d'una mateixa partida, a través d'Internet.

Joystick: Un joystick, també conegut com a palanca de control, és un dispositiu de control que pivota en una base i comunica al sistema varis tipus d'informació, com l'angle i la direcció a la que està inclinat.

Jugador: Un jugador és una persona física que juga a un joc determinat.

Personatge: En els videojocs, un personatge és una representació virtual del jugador que s'utilitza en el joc.

SDK (Software Development Kit): Un SDK és un conjunt d'eines que s'utilitzen a l'hora de desenvolupar un software per un sistema concret. Entre aquestes eines es poden incloure l'API del sistema, suport de detecció d'errors de programació, un entorn de programació, codis d'exemple o documentació per ajudar a programar.

XBLA (Xbox Live Arcade): És un sistema desenvolupat per Microsoft que serveix per distribuir jocs digitals, generalment més petits i barats que els jocs comercials que es venen en format físic.

XBLIG (Xbox Live Indie Games): Els jocs XBLIG són videojocs independents desenvolupats per una o més persones per a la consola Xbox 360. Estan escrits en XNA i es poden penjar en el Bazar a través d'App Hub. Per poder-ho fer, és necessari que els desenvolupadors siguin membres creadors d'App Hub.

Xbox Live: Xbox Live és el servei online de Microsoft que permet que els jugadors de Xbox 360 puguin jugar entre ells mitjançant Internet. També permet l'accés al servei de distribució de contingut digital. Hi ha dos tipus de comptes Xbox Live: Silver, que permet l'accés al Bazar i a les actualitzacions dels jocs, i Gold, que a més de les característiques del compte Silver també permet que els jugadors puguin competir online. Mentre el compte Silver és gratuït, el compte Gold funciona a través de subscripcions de pagament de varis mesos.

XNA (XNA's Not Acronymed): XNA és el conjunt d'eines de Microsoft que faciliten el desenvolupament de jocs per Windows i Xbox 360. A l'apartat 3.2 de la memòria s'aprofundeix més en aquestes eines.

2.2 Parts interessades

En aquest apartat es mostren la composició dels equips implicats en el projecte, així com els rols i responsabilitats de cadascú.

2.2.1 Stakeholders

Nom	Descripció	Responsabilitat
Xavier Verge	Director del projecte.	Supervisa el projecte.
Miquel Sitges	Responsable.	Defineix els requisits i les funcionalitats.

2.2.2 Perfils d'usuaris

Nom	Descripció	Responsabilitat
Miquel Sitges	Desenvolupador.	Programa el joc i s'encarrega de la distribució.
Betatesters	Betatester.	Prova l'aplicació durant la versió beta.
Jugadors	Jugador.	Prova l'aplicació final.

2.2.3 Equip del projecte

Nom	Descripció	Responsabilitat
Miquel Sitges	Cap del projecte.	Defineix, gestiona i controla el projecte.
Miquel Sitges	Analista.	Realitza l'estudi de viabilitat i la planificació.
Miquel Sitges	Programador.	Desenvolupa el joc.
Betatesters	Betatesters.	Proven el joc i retornen un feedback.
Xavier Verge	Director del projecte.	Supervisa el projecte i actua com a <i>stakeholder</i> .

2.3 Requisits del sistema

Amb els requisits del sistema es mostren els punts necessaris que s'han de dur a terme per aconseguir que el videojoc tingui un bon resultat final.

2.3.1 Requisits funcionals (RF)

1. Creació del joc en XNA: Per a poder desenvolupar un videojoc indie (XBLIG) per a Xbox 360, és un requisit indispensable desenvolupar utilitzant les eines XNA.

2. Lectura i interpretació de l'input dels comandaments dels jugadors: Els jugadors es comuniquen amb la consola i el joc mitjançant els botons i les palanques que tenen els comandaments de la consola. Per tant, quan un jugador genera un *input* amb el comandament, el joc ha de llegir-lo i interpretar-lo.
3. Creació de menús funcionals: El joc ha de tenir varis menús en els que s'expliqui el funcionament del joc, permeti modificar les característiques de la partida i finalment porti els jugadors a la partida.
4. Creació de varis espais 3D per on es puguin moure els personatges: Els personatges s'han de poder moure en escenaris 3D generats per la consola, de forma que estigui clar quina és la direcció que han de seguir per aconseguir arribar a la meta.
5. Vinculació de cada jugador a un avatar en el joc: Cada jugador ha d'utilitzar un compte Xbox diferent per poder jugar. El joc utilitza l'avatar de cada compte Xbox per a representar al jugador.
6. Introducció i implementació de varis moviments per a cada personatge: Cada acció ha de tenir un moviment diferent perquè el jugador vegi de forma gràfica que l'acció s'ha realitzat correctament.
7. Crear un sistema de càmeres per al joc multijugador: Quan hi hagi varis jugadors jugant alhora, la pantalla s'ha de dividir de forma que cada un tingui una càmera que el segueixi de forma individual.
8. Implementació d'una intel·ligència artificial senzilla: Si no hi ha els suficients jugadors per a completar una partida, el joc ha de portar els personatges restants com si es tractessin de persones reals.
9. Creació de varis nivells de resposta per als personatges controlats per a la IA: Els jugadors poden personalitzar el nivell de dificultat dels personatges que controla la intel·ligència artificial.
10. Creació d'un sistema d'atacs i defenses pels personatges: Per a que un jugador pugui avançar-se als seus contrincants, podrà disposar de varis atacs i defenses que l'ajudaran.

11. Creació d'una tenda on comprar els atacs i les defenses: Els atacs i les defenses s'han de comprar a través de la tenda.
12. Creació d'un sistema d'energia vinculat als atacs, les defenses i al moviment dels personatges: Els personatges tindran una energia finita, que es gastarà quan es moguin per l'escenari o quan els ataquin. Només quan els personatges estiguin quiets o utilitzin un objecte de defensa podran recuperar energia. Si l'energia s'esgota, el personatge queda immobilitzat durant uns segons.
13. Omplir els escenaris amb obstacles: Els escenaris han de tenir objectes tridimensionals que obstaculitzin el pas dels personatges i que millorin l'aspecte de l'escenari.
14. Creació d'una zona per a respondre la pregunta formulada: Els personatges han de contestar les preguntes a la zona de respostes, on han d'escollir la resposta correcta d'entre les que se'ls ofereix.
15. Creació d'un sistema de puntuació: El joc ha de calcular la puntuació de cada personatge, tenint en compte si ha contestat correctament la pregunta i en quina posició respecte els demés rivals ha contestat.
16. Creació d'un sistema d'estadístiques de joc: El joc ha de guardar un historial dels cops que ha guanyat cada personatge i en quines posicions ha quedat.
17. Implementació d'un menú contextual de pausa: Els jugadors han de poder pausar el joc en qualsevol moment mentre estiguin jugant. El joc s'ha d'aturar i mostrar un menú per poder canviar algunes de les opcions o continuar la partida.
18. Introducció de músiques i efectes de so: El joc ha de tenir algunes músiques i efectes de so per acompanyar al gameplay.
19. Introducció de varies categories per a les preguntes: El joc ha de suportar varies categories de preguntes, amb varies preguntes per a cada categoria. Aquestes preguntes no s'han de repetir dins d'una mateixa partida.

20. Creació d'una interfície gràfica per a la partida: Per a fer més intuïtiva la partida, el jugador ha de veure de forma gràfica quanta energia li queda, així com la quantitat d'objectes d'atac i defensa que li queda per utilitzar.

2.3.2 Requisits no funcionals (RNF)

1. Tenir un compte de desenvolupador App Hub actiu: Per a poder desenvolupar un joc XBLIG per a Xbox, és necessari tenir un compte d'App Hub actiu.
2. Compliment de l'autorització de publicació de software de Microsoft: El joc ha de complir un seguit de normes per a poder ser aprovat i publicat al Bazar de Microsoft.
3. Compliment de les normes XNA: El joc ha de complir un seguit de normes per a poder ser aprovat i publicat al Bazar de Microsoft.
4. Compliment de les normes App Hub: El joc ha de complir un seguit de normes per a poder ser aprovat i publicat al Bazar de Microsoft.

2.3.3 Priorització dels requisits

Priorització dels requisits funcionals:

	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9	RF10
Essencial	X	X	X	X		X	X	X		X
Condicional					X					
Opcional									X	

	RF11	RF12	RF13	RF14	RF15	RF16	RF17	RF18	RF19	RF20
Essencial	X	X	X							X
Condicional				X	X		X	X		
Opcional						X			X	

Priorització dels requisits no funcionals:

	RNF1	RNF2	RNF3	RNF4
Essencial	X			
Condicional		X	X	X
Opcional				

2.3.4 Relació dels requisits amb els objectius

	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	O13	O14
RF1	X	X	X	X	X	X	X	X		X	X	X	X	X
RF2	X					X				X	X			X
RF3										X		X		
RF4	X							X						
RF5	X	X			X	X	X				X			
RF6						X	X				X			
RF7	X	X	X											
RF8	X				X	X	X				X			X
RF9	X				X						X			
RF10	X				X		X				X	X		X
RF11	X				X					X		X		
RF12	X	X			X		X				X			X
RF13	X				X			X						
RF14	X	X			X			X					X	
RF15	X	X											X	
RF16	X	X											X	X
RF17	X	X								X				
RF18	X			X										
RF19	X								X				X	
RF20	X		X								X			X
RNF1	X	X	X	X	X	X	X	X		X	X	X	X	X
RNF2	X	X	X	X	X	X	X	X		X	X	X	X	X
RNF3	X	X	X	X	X	X	X	X		X	X	X	X	X
RNF4	X	X	X	X	X	X	X	X		X	X	X	X	X

2.4 Alternatives

L'objectiu principal del projecte, desenvolupar un videojoc, es pot aconseguir de moltes formes diferents. És per això que s'ha de pensar en quines són les millors alternatives possibles per aconseguir-ho i escollir la més viable de totes tenint en compte els avantatges i els inconvenients de cadascuna.

En cada possible alternativa s'hi poden trobar varis elements necessaris per dur a terme el desenvolupament. Per exemple, en totes les alternatives necessitarem com a mínim un ordinador, ja que es necessita per la programació, així com el software bàsic per a poder desenvolupar el projecte (sistema operatiu, programes de desenvolupament, programes de disseny gràfic, etc.). Un altre element comú, en aquest cas per a totes les alternatives en les que es desenvolupa el joc per a Xbox 360, és la mateixa consola. Tots aquests elements no han estat tinguts en

compte a l'hora de valorar els costos de cada alternativa, ja que són elements comuns que totes necessiten.

2.4.1 Alternativa 1

Joc desenvolupat en XNA per a Xbox, permet jugar varies persones al mateix instant de forma local. Utilitza les 2D a nivell gràfic. Un projecte més senzill de desenvolupar degut a la implementació gràfica, i per tant més ràpid de fer.

Característiques:

- Joc desenvolupat per a Xbox 360.
- Enfocat a XBLIG.
- Multijugador local.
- Gràfics en 2D.
- Preguntes en XML.
- No permet l'actualització del qüestionari de preguntes per persones diferents al desenvolupador.

Costos:

- Dificultat baixa.
- Poc costos temporalment.
- Poca inversió necessària.

2.4.2 Alternativa 2

Joc desenvolupat en XNA per a Xbox, permet jugar varies persones al mateix instant de forma local. Utilitza les 3D a nivell gràfic. Un projecte més vistós i que dona més joc al jugador.

Característiques:

- Joc desenvolupat per a Xbox 360.
- Enfocat a XBLIG.
- Multijugador local.
- Gràfics en 3D.
- Permet utilitzar els avatars de la consola Xbox com a personatges de joc.
- Preguntes en XML.
- No permet l'actualització del qüestionari de preguntes per persones diferents al desenvolupador.

Costos:

- Dificultat mitja.

- Temporalment assequible dins dels límits.
- Poca inversió necessària.

2.4.3 Alternativa 3

Joc desenvolupat en XNA per a Xbox, permet jugar vàries persones al mateix instant de forma local o online. Utilitza les 3D a nivell gràfic. Un projecte molt més complet, que necessita més temps per a desenvolupar-se.

Característiques:

- Joc desenvolupat per a Xbox 360.
- Enfocat a XBLIG.
- Multijugador local i online.
- Gràfics 3D.
- Permet utilitzar els avatars de la consola Xbox com a personatges de joc.
- Preguntes en XML.
- No permet l'actualització del qüestionari de preguntes per a persones diferents al desenvolupador.

Costos:

- Dificultat mitja/elevada.
- Cost temporal difícil de dur a terme dins dels límits fixats.
- Inversió necessària mitja (connexions Gold per a les consoles Xbox 360).

2.4.4 Alternativa 4

Joc desenvolupat en el SDK oficial de Xbox, el que permetria tenir menys restriccions a l'hora de desenvolupar, com per exemple el pes final del joc, així com la possibilitat de poder publicar-lo com a XBLA. Permet jugar vàries persones al mateix instant de forma local o online. Utilitza les 3D a nivell gràfic.

Característiques:

- Joc desenvolupat per a Xbox 360.
- Enfocat a XBLA.
- Multijugador local i online.
- Gràfics 3D.
- Permet utilitzar els avatars de la consola Xbox com a personatges de joc.
- Preguntes en XML
- No permet l'actualització del qüestionari de preguntes per a persones diferents al desenvolupador.

Costos:

- Dificultat alta.
- Cost temporal no assequible dins els terminis marcats.
- Inversió necessària molt elevada (SDK oficial).

2.4.5 Alternativa 5

Joc desenvolupat en C++ per a PC. De forma local, únicament permet el joc individual, tot i que es pot jugar en multijugador de forma online. Utilitza les 3D a nivell gràfic. Sense les restriccions de desenvolupar per a Xbox, tenim més llibertat a l'hora de desenvolupar el joc.

Característiques:

- Joc desenvolupat per a PC.
- Multijugador únicament online.
- Gràfics 3D.
- Preguntes en XML.
- Permet l'actualització dels arxius XML mitjançant un formulari web, accessible a tothom.

Costos:

- Dificultat mitja/alta.
- Cost temporal no assequible dins els terminis marcats.
- Inversió necessària molt elevada (servidors i manteniment).

2.4.6 Solució proposada

Idealment, les alternatives 4 i 5 són les més completes i les que donen més llibertat a l'hora de desenvolupar el joc. Aquí l'alternativa 5 és molt interessant degut a la possibilitat d'una actualització senzilla del qüestionari de preguntes, mitjançant una pàgina web. Les contres d'aquestes alternatives són els costos massa elevats: l'alternativa 4 suposa l'adquisició del SDK oficial (aproximadament uns 20.000\$), mentre que amb la 5 s'han de costejar els servidors que permeten les competicions online, així com el servidor web d'actualització del qüestionari. A més, existeixen altres contres, com el fet de formar una companyia de videojocs per accedir a la compra del SDK de l'alternativa 4 o el d'haver de dissenyar els personatges de forma manual a l'alternativa 5, al no disposar dels avatars de Xbox, el que retardaria el desenvolupament.

L'alternativa 1 és la més ràpida i fàcil de desenvolupar. Per tant, es guanya força amb el factor temps, tot i que el joc no és tant espectacular a nivell visual i fins i tot podria quedar *antic* pel fet d'utilitzar les 2D en comptes de les 3D.

Les alternatives 2 i 3 són per tant les més assequibles i balancejades a nivell de qualitat, cost i esforç. L'única diferència és que l'alternativa 3 permet multijugador online, mentre que l'alternativa 2 només té multijugador local. A canvi, l'alternativa 3 necessita molt més temps per de desenvolupament.

Com que una part del temps del projecte està destinat a aprendre a programar per a Xbox, el temps disponible per a desenvolupar-lo no és suficient per dur a terme l'alternativa 3. És per això que en aquest cas, la millor alternativa tenint en compte el temps i els recursos disponibles és l'alternativa 2.

2.5 Planificació

El projecte ha tingut sempre dos objectius clars. El primer, aplicar els coneixements que s'han après durant la carrera en el desenvolupament d'un videojoc. El segon, aprofitar la temàtica del projecte per fer una formació en la programació per a consoles Xbox.

Per tant, i com que els coneixements sobre la programació en consoles són molt bàsics, les primeres setmanes s'han d'invertir en investigació tant del codi com de les eines que es faran servir pel desenvolupament del projecte (C# mitjançant les llibreries XNA). Paral·lelament, també s'ha de dissenyar el tipus de joc que es portarà a terme, així com les seves mecàniques i necessitats. És important tenir clar què tindrà el joc i què necessita, per dedicar el temps just a la investigació i passar tant aviat com sigui possible al desenvolupament.

Un cop apresos els conceptes bàsics de la programació amb gràfics 3D en Xbox, i tenint en compte les necessitats del videojoc que s'han decidit, es passa a investigar les característiques més importants i a desenvolupar-les per separat, per treballar amb un codi més simplificat i saber exactament com funciona cada part. Totes aquestes proves es reutilitzaran més endavant un cop s'integrin en el projecte final.

A mida que es van tenint les proves més bàsiques per a poder muntar la base del videojoc, es decideix el model de desenvolupament del projecte. Es marquen una sèrie de fites al calendari, on en cada una d'elles s'han d'aconseguir una sèrie d'objectius.

A cada fita es poden revisar els objectius de la propera. Per tant, el model de desenvolupament és iteratiu, tot i que s'intenten tenir amb antelació el màxim de fites possibles pensades. Així doncs, també es busca que el desenvolupament sigui com un model clàssic de cascada tant com sigui possible.

El projecte es va començar a principis d'octubre de 2011 i es va entregar a mitjans de juny de 2012.

2.5.1 Tasques i fites

Fita 1

Objectius de la fita:

- Brainstorming i disseny del joc. (3h)
- Prova d'inclusió d'avatars al projecte. (2h)

Dates: 3 d'octubre – 20 d'octubre.

Hores dedicades: 5 hores.

Fita 2

Objectius de la fita:

- Prova d'animacions múltiples de l'avatar. (4h)
- Prova d'animació personalitzada de l'avatar. (15h)
- Prova de moviment de l'avatar per l'escenari. (6h)
- Prova de lectura d'un arxiu XML. (2h)
- Prova de lectura de l'input del joystick del comandament. (2h)
- Prova partició de pantalla. (7h)

Dates: 21 d'octubre – 15 de novembre

Hores dedicades: 36 hores

Fita 3: Alpha 1

Objectius de la fita:

- Primeres versions alpha mesclant les proves de les fites anteriors. (10h)
- Marcats els límits físics de l'escenari. (4h)
- Partició dinàmica de la pantalla segons el nombre de jugadors. (8h)
- Ajust del punt de visió de les càmeres del multijugador. (22h)
- Codi principal dividit i estructurat en classes. (5h)
- Implementació dels menús i els salts entre menús. (3h)

Dates: 16 de novembre – 20 de desembre

Hores dedicades: 52 hores

Fita 4: Alpha 2

Objectius de la fita:

- Inicialització de les sessions dels jugadors en el joc. (5h)
- Implementació de la pausa durant la partida. (7h)
- Inclusió de la tenda d'objectes. (4h)
- Col·lisions entre jugadors. (18h)
- Millorar el sistema per controlar l'input de cada jugador. (17h)
- Millorar la gestió de les animacions de l'avatar. (4h)
- Crear un algorisme d'aleatorietat per les animacions "stand" de l'avatar. (5h)
- Afegir algunes animacions personalitzades i sincronització de les animacions múltiples. (5h)
- Correcció de bugs. (3h)

Dates: 21 de desembre – 13 de gener

Hores dedicades: 68 hores

Fita 5: Alpha 3

Objectius de la fita:

- Introducció d'objectes a l'escenari. (19h)
- Implementar un sistema de col·lisions entre objectes. (25h)
- Crear 3 escenaris que tinguin una ordenació d'objectes diferents. (6h)
- Integrar la prova de lectura d'arxius XML i ampliar-la a 3 preguntes aleatòries per joc, sense repetició dins de la mateixa partida. (4h)
- Crear la zona de respostes. (7h)
- Implementar un sistema de respostes dels personatges. (4h)
- Correcció de bugs. (4h)

Dates: 14 de gener – 30 de gener

Hores dedicades: 69 hores

Fita 6: Alpha 4

Objectius de la fita:

- Mostrar el text de les preguntes i les respostes a l'escenari. (3h)
- Crear un cicle al menú de la tenda per a que entrin tots els jugadors que hagin iniciat la sessió. (2h)
- Crear un inventari pels objectes comprats i un mètode per utilitzar-los durant la partida. (3h)
- Creació de menús dinàmics abans de la partida i modificar els que hi ha per poder ajustar la categoria de preguntes i la dificultat de la IA. (4h)
- Creació del submenú opcions dins del menú pausa. (2h)

- Evitar que els jugadors puguin iniciar o tancar sessió a les pantalles de tenda i de joc. (3h)
- Correcció de bugs. (5h)

Dates: 31 de gener – 13 de febrer

Hores dedicades: 22 hores

Fita 7: Alpha 5

Objectius de la fita:

- Començar a implementar la IA:
 - o La IA ha d'escollir una resposta més o menys encertada segons el nivell que hi hagi seleccionat. (2h)
 - o La IA ha d'escollir el camí més o menys correcte fins arribar al final de l'escenari segons el nivell que hi hagi seleccionat. (16h)
- Implementar les animacions dels avatars controlats per la IA. (6h)
- Correcció de bugs. (3h)

Dates: 14 de febrer – 27 de febrer

Hores dedicades: 27 hores

Fita 8: Alpha 6

Objectius de la fita:

- Implementar el sistema d'energia de cada jugador. (8h)
- Crear un element gràfic que mostri l'energia restant del jugador. (2h)
- Implementar el sistema de bonificació en punts d'atac segons l'ordre en que contesti cada personatge. (5h)
- Implementar el sistema d'atac i defensa dels jugadors reals. (13h)
- Correcció de bugs. (2h)

Dates: 28 de febrer – 12 de març

Hores dedicades: 30 hores

Fita 9: Alpha 7

Objectius de la fita:

- Implementar el sistema d'atac i defensa dels personatges controlats per la IA. (8h)
- Implementar la compra d'objectes automàtica per als personatges controlats per la IA. (3h)
- Implementar el sistema de bonificació en punts de defensa segons siguin atacats els personatges. (3h)
- Començar a implementar el sistema de música i efectes de so. (4h)
- Reestructuració del sistema d'animacions i estats de la IA. (14h)
- Correcció de bugs. (5h)

Dates: 13 de març – 26 de març

Hores dedicades: 37 hores

Fita 10: Beta 1

Objectius de la fita:

- Acabar la implementació del sistema de música i efectes de so. (2h)
- Crear una interfície gràfica perquè es mostri durant la partida. (10h)
- Millorar el disseny dels menús. (8h)
- Implementar una pantalla que mostri els guanyadors de cada ronda i la seva puntuació. (3h)
- Correcció de bugs. (2h)

Dates: 27 de març – 11 d'abril

Hores dedicades: 25 hores

Fita 11: Beta 2

Objectius de la fita:

- Implementar una pantalla que mostri els guanyadors de cada joc i la seva puntuació. (5h)
- Afegir un opció que mostri les instruccions del joc. (2h)
- Escollir i afegir les preguntes als arxius XML. (2h)
- Afegir un element visual per saber quan un jugador ha contestat una resposta. (3h)
- Correcció de bugs. (7h)

Dates: 12 d'abril – 7 de maig

Hores dedicades: 19 hores

Fita 12: Beta 3

Objectius de la fita:

- Betatesting. (3h)
- Correcció de bugs. (7h)

Dates: 8 de maig – 11 de maig

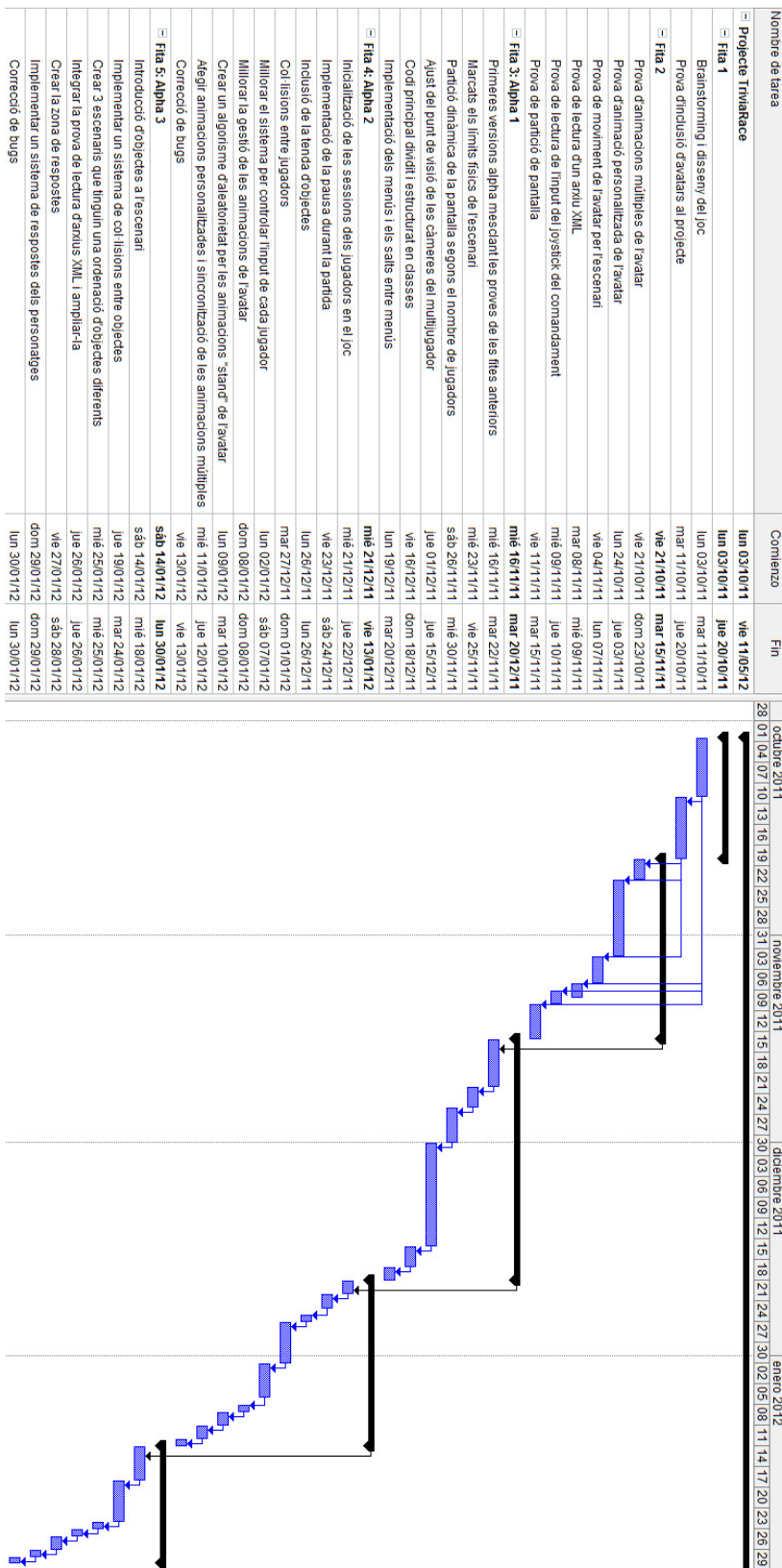
Hores dedicades: 10 hores

Total d'hores dedicades: 400 hores.

Fent un recompte de les hores dedicades, es pot veure que supera les hores aproximades que hauria de durar el desenvolupament d'un projecte de final de carrera. Tot i això, queda clar que per aconseguir desenvolupar un videojoc d'aquestes característiques partint d'uns coneixements bàsics en XNA és el mínim

necessari d'hores que s'hi ha de dedicar per poder donar com a finalitzat el projecte i complir els objectius principals.

2.5.2 Diagrama de Gantt



Nombre de tarea	Comienzo	Fin
Fita 6: Alpha 4	mar 31/01/12	lun 13/02/12
Mostrar el text de les preguntes i les respostes	mar 31/01/12	mié 01/02/12
Crear un ciclo al menú de la tienda	jue 02/02/12	jue 02/02/12
Crear un inventari i un mètode per utilitzar els objectes	vie 03/02/12	sáb 04/02/12
Creació i modificació de menús dinàmics	dom 05/02/12	mar 07/02/12
Creació del submenú opcions dins del menú pausa	mié 08/02/12	mié 08/02/12
Evitar que els jugadors puguin iniciar o lancar sessió a les pantalles de tenda i de joc	jue 09/02/12	vie 10/02/12
Correcció de bugs	sáb 11/02/12	lun 13/02/12
Fita 7: Alpha 5	mar 14/02/12	lun 27/02/12
Començar a implementar la IA	mar 14/02/12	mié 22/02/12
La IA ha de seleccionar una resposta segons el seu nivell	mar 14/02/12	mar 14/02/12
La IA ha de seleccionar el canvi segons el seu nivell	mié 15/02/12	mié 22/02/12
Implementar les animacions dels avatars controlats per la IA	jue 23/02/12	sáb 25/02/12
Correcció de bugs	dom 26/02/12	lun 27/02/12
Fita 8: Alpha 6	mar 28/02/12	lun 12/03/12
Implementar el sistema d'energia de cada jugador	mar 28/02/12	vie 02/03/12
Crear un element gràfic que mostri l'energia	sáb 03/03/12	sáb 03/03/12
Implementar el sistema de bonificació en punts d'atac	dom 04/03/12	lun 05/03/12
Implementar el sistema d'atac i defensa dels jugadors reals	mar 06/03/12	dom 11/03/12
Correcció de bugs	lun 12/03/12	lun 12/03/12
Fita 9: Alpha 7	mar 13/03/12	lun 26/03/12
Implementar el sistema d'atac i defensa de la IA	mar 13/03/12	jue 15/03/12
Implementar la compra d'objectes automàtica per a la IA	vie 16/03/12	vie 16/03/12
Implementar el sistema de bonificació en punts de defensa	sáb 17/03/12	sáb 17/03/12
Començar a implementar el sistema de música i so	dom 18/03/12	lun 19/03/12
Reestructuració del sistema d'animacions i estats de la IA	mar 20/03/12	sáb 24/03/12
Correcció de bugs	dom 25/03/12	lun 26/03/12
Fita 10: Beta 1	mar 27/03/12	mié 11/04/12
Acabar la implementació del sistema de música i so	mar 27/03/12	mar 27/03/12
Crear una interfície gràfica	mié 28/03/12	mar 03/04/12
Millorar el disseny dels menús	mié 04/04/12	dom 08/04/12
Implementar una pantalla que mostri els guanyadors de la ronda	lun 09/04/12	mar 10/04/12
Correcció de bugs	mié 11/04/12	mié 11/04/12
Fita 11: Beta 2	 jue 12/04/12	 lun 07/05/12
Implementar una pantalla que mostri els guanyadors de cada joc	jue 12/04/12	lun 16/04/12
Afegir una opció que mostri les instruccions del joc	vie 27/04/12	vie 27/04/12
Afegir i afegir les preguntes als arxius XML	sáb 28/04/12	sáb 28/04/12
Correcció de bugs	dom 29/04/12	mar 01/05/12
Fita 12: Beta 3	mié 02/05/12	 lun 07/05/12
Belatesting	mar 08/05/12	 vie 11/05/12
Correcció de bugs	mar 08/05/12	mar 08/05/12
Correcció de bugs	mié 09/05/12	vie 11/05/12

2.6 Avaluació dels riscos

És molt normal que durant un projecte de desenvolupament de software sorgeixin alguns imprevistos, però tot i això s'ha de mirar sempre d'evitar-los tant com es pugui. És possible que un imprevist comporti un risc molt important per al projecte, podent arribar a ser fatal. A més, aquest projecte parteix de la base que els coneixements sobre la matèria són molt escassos i s'ha d'aprendre prèviament, podent comportar forces contratemps. És per això que s'han de preveure tant com sigui possible, calculant el risc que comportaria cada contratemps i evitant una situació catastròfica.

2.6.1 Identificació de riscos

R1. Planificació temporal optimista: Pot sorgir durant la fase de l'estudi de viabilitat. No s'acaba en la data prevista.

R2. Manca alguna tasca necessària: Pot sorgir durant la fase de l'estudi de viabilitat i disseny del joc. No es compleixen els objectius.

R3. Canvi de requisits: Pot sorgir durant la fase de l'estudi de viabilitat, el disseny del joc i el desenvolupament. Endarreriment en el desenvolupament.

R4. Fallada de les eines o les llicències: Pot sorgir durant qualsevol fase. Endarreriment de la planificació depenent dels components o les llicències fallats.

R5. Fase de test inadequada: Pot sorgir durant la fase de disseny de les proves i la fase de test. Producte mal acabat i poca qualitat.

R6. Incompliment de normes legals: Pot sorgir durant qualsevol fase. Problemes legals.

R7. Abandonament del projecte abans de la finalització: Pot sorgir durant qualsevol fase. Pèrdues econòmiques i de motivació.

2.6.2 Catalogació dels riscos

	Probabilitat	Impacte
R1	Alta	Crític
R2	Alta	Crític
R3	Mitjana	Marginal
R4	Mitjana	Crític
R5	Baixa	Crític
R6	Baixa	Crític
R7	Mitjana	Catastròfic

2.6.3 Pla de contingència

	Solució
R1	Ajornar alguna funcionalitat, aplaçar la data de finalització.
R2	Revisar l'estudi de viabilitat, modificació de la previsió.
R3	Ampliar les hores de treball.
R4	Ajornar alguna funcionalitat, comprar una eina o llicència nova.
R5	Revisar l'anàlisi de test, proposar ajuda a la comunitat desenvolupadora.
R6	Revisar les normes i la legislació.
R7	No té solució.

2.7 Pressupost

2.7.1 Costos de personal

Recursos humans	Valoració
Cap del projecte. (CP)	50€/h
Analista. (An)	25€/h
Programador. (Pr)	15€/h
Betatesters. (Bt)	5€/h

2.7.2 Costos dels recursos

- 1 PC amb Windows 7: 700€
- 1 Xbox 360: 249,95€
- 1 Microsoft XNA Game Studio 4.0: 0€
- 1 Microsoft Visual Studio 2010 Express: 0€
- 2 comptes XNA Creator's Club: 0€
- 1 GIMP: 0€
- 1 Blender: 0€

2.7.3 Costos d'amortització

	Cost amortització	Cost unitari	Període amortització	Període utilització
PC	175€	700€	36 mesos	9 mesos
Xbox 360	62,49€	249,95€	36 mesos	9 mesos
XNA 4.0	0€	0€	24 mesos	9 mesos
Visual Studio	0€	0€	36 mesos	9 mesos
XNA Creator's Club	0€	0€	12 mesos	9 mesos
GIMP	0€	0€	36 mesos	2 mesos
Blender	0€	0€	12 mesos	2 mesos

2.8 Conclusions de l'estudi de viabilitat

Un cop definit l'estudi de viabilitat, queda per determinar si el projecte és viable o és necessari pensar en una alternativa diferent.

En cas de continuar amb aquest projecte, el joc podria obrir-se un forat dins del mercat ja que no hi ha cap joc semblant entre els jocs educatius. A més ja s'ha buscat una alternativa que tingués en compte les despeses econòmiques, i per tant la inversió inicial no és molt elevada. Això es pot aconseguir gràcies a les llicències d'estudiant que permeten utilitzar la majoria de programes de forma gratuïta i als programes de codi lliure que no necessiten llicència.

La part més negativa de l'estudi de viabilitat és la quantitat d'hores necessàries per dur a terme el projecte. Les hores de treball superen les que s'haurien de dedicar en un projecte de final de carrera, però per un joc d'aquest tipus són el mínim d'hores necessàries per a tenir uns resultats que permetin donar el projecte com a finalitzat. Aquesta diferència d'hores sumada al fet que al començar el projecte els coneixements de programació per a Xbox 360 són molt bàsics pot provocar un endarreriment dins de la planificació.

Tot i això el projecte és molt motivador i, si es finalitzés, l'experiència i els coneixements adquirits són molt interessants a nivell personal. Per tant, es pot considerar que els riscos són assumibles i fàcils de fer-ne un seguiment per detectar amb temps suficient si la planificació es desvia de forma excessiva.

Si tenim en compte els beneficis aportats al dur a terme el projecte en contra dels possibles inconvenients, es pot concloure que el projecte és viable.

3. Fonaments teòrics

3.1 Xbox 360

La Xbox 360 és la segona consola de videojocs desenvolupada per Microsoft i una de les principals eines que s'utilitzen en aquest projecte. Competeix en el mercat juntament amb les consoles de Sony i Nintendo: la PlayStation 3 i la Nintendo Wii.

La consola es va llançar als mercats nord-americà, europeu i japonès de forma simultània a finals del 2005, i posteriorment en més de 35 països. A l'any 2010, Microsoft va redissenyar la Xbox 360, reduint les mides i el pes respecte la versió antiga, i afegint alguns serveis de forma integrada com l'antena Wi-Fi, que en la seva versió original s'havia d'adquirir apart, o el disc dur, que passa a ser intern i de capacitats més elevades que els anteriors. Aquesta nova versió de la consola es va anomenar Xbox 360 S. Microsoft ha anunciat que a dia 30 de maig de 2012 s'han venut més de 67 milions de consoles en tot el món.

Tot i haver hagut dues versions diferenciades de la consola, dins de cada una s'han donat variacions en les prestacions. Per exemple, en la Xbox 360 original hi ha hagut varis models disponibles, com per exemple el model Core que no inclou disc dur (obligant al client a comprar el disc dur o la targeta de memòria de forma separada), el model Premium que inclou disc dur (de 20 o 60 GB de capacitat) i el model Elite que inclou disc dur de 120 GB i cable HDMI. La principal diferència entre els models S ve marcada per la capacitat del disc dur. Els models més assequibles incorporen una memòria de 4 GB de capacitat, mentre que els demés models inclouen un disc dur de 250 i 320 GB de capacitat.



Figura 1: Xbox 360 Elite (esquerra), Xbox 360 S (dreta)

Les característiques tècniques d'ambdues versions són les següents:

CPU: PowerPC Tri-Core Xenon 3,2 GHz

Memòria: 512 MB de RAM GDDR3 a una freqüència de 700 MHz.

GPU: ATI Xenos 500 MHz

Formats suportats: DVD, CD, HD-DVD (sense suport), descàrrega.

Emmagatzematge:

- Discs durs: 20 GB, 60 GB, 120 GB, 250 GB, 320 GB.
- Targetes de memòria: 64 MB, 256 MB, 512 MB.
- Memòries integrades: 256 MB, 512 MB, 4 GB.
- Dispositius USB: de 1 GB fins a 16 GB.
- Núvol: 512 MB.

Formats de vídeo: Compost, S-Video, RGB SCART, components, D-Terminal, VGA, HDMI.

Format de so: Estèreo analògic, estèreo LPCM, Dolby Digital 5.1, Dolby Digital WMA pro.

Controladors d'entrada (fins a 4 de forma simultània): Controlador Xbox 360 cablejat, controlador Xbox 360 inalàmbric (2,4 GHz), Kinect, entre d'altres.

Connectivitat:

- Models originals: 2,4 GHz wireless, 3x USB 2.0, receptor IR, 100 Mbit Ethernet. Accessoris: Wi-Fi 802.11a/b/g, Wi-Fi 802.11a/b/g/n.
- Models S: 2,4 GHz wireless, 5x USB 2.0, receptor IR, 100 Mbit Ethernet, Wi-Fi 802.11b/g/n, port AUX.

Serveis online: Xbox Live.

La Xbox 360 basa el seu èxit en diversos pilars, sent evidentment el més important els seus jocs comercials. Tot i això, també aposta fortament pel contingut online, a través de la seva plataforma Xbox Live. Aquesta plataforma permet als jugadors competir entre ells amb els jocs que tinguin, parlar entre ells mentre estan jugant (ja sigui en el mateix joc o en diferents), tenir una llista de fins a 100 amics (amb els que es poden comparar les puntuacions), comprar i descarregar jocs, entre altres serveis.

Tot i que la majoria de serveis son gratuïts, per poder jugar de forma online amb més jugadors és necessari que cada un d'ells hagi comprat una subscripció de pagament, anomenada subscripció Gold, que permet accedir a la majoria dels serveis de pagament durant un temps limitat (segons el tipus de subscripció adquirida).

3.2 Microsoft XNA

Microsoft XNA és un recull d'eines per a desenvolupar software, orientades sobretot al desenvolupament de videojocs. El nom XNA significa *XNA's Not Acronymed* (XNA no és un acrònim). Actualment permet desenvolupar jocs per a tres plataformes: Microsoft Windows, Xbox 360 i Windows Phone.

XNA està inclòs en les seccions de Game Development Sections de Microsoft i inclou el SDK de Xbox estàndard i el conjunt de programes XNA Game Studio. El componen:

- XNA Framework: Està basat en .NET Compact Framework 2.0 per a Xbox 360 i .NET Framework 2.0 en Windows. Inclou un gran conjunt de llibreries específiques per al desenvolupament de videojocs i busca la reutilització màxima de codi entre les plataformes. Com els jocs s'executen en temps real, només funcionen en plataformes que suportin XNA Framework. Tot i que amb XNA es pot programar en qualsevol llenguatge .NET, només està suportat de forma oficial el llenguatge C# per als programes XNA Game Studio Express, i totes les versions de Visual Studio 2008 i 2010.

XNA Framework també ajuda als desenvolupadors a concentrar-se més en el contingut i l'experiència de joc, encarregant-se dels detalls de baix nivell i de les diferències que puguin tenir les plataformes per a les que es pot desenvolupar. Integra varies eines i dona suport a jocs 2D i 3D, permetent l'ús dels controladors Xbox 360.

Per a poder desenvolupar per a Xbox 360, es necessita a més una subscripció al Microsoft XNA Creator's Club, que costa 99\$ o 75€ per any. En el cas dels estudiants, Microsoft dona llicències anuals per estudiants de forma gratuïta, a través de la pàgina DreamSpark o del programa MSDNAA.

- XNA Build: És un conjunt d'eines per administrar, mantenir, depurar i optimitzar el codi. També s'encarrega de les dependències de dades, analitzant el contingut del projecte i trobant tot allò que no s'utilitza.
- XNA Game Studio: És l'entorn de desenvolupament que s'utilitza per a escriure els jocs. Des del llançament de la seva primera versió l'11 de desembre de 2006, Microsoft ha llançat un total de 5 versions:

- XNA Game Studio Express: La primera versió del XNA Game Studio anava orientada per a estudiants i desenvolupadors independents. Inclou un paquet bàsic de desenvolupament ràpid per a gèneres de jocs específics. Tot i que al principi no suportava la compartició de jocs pre-compilats de Xbox 360 a altres membres del XNA Creator's Club, es va afegir aquest servei mitjançant una actualització anomenada "1.0 Refresh".
 - XNA Game Studio 2.0: Aquesta versió destaca per funcionar en qualsevol versió de Visual Studio 2005, poder utilitzar l'API de Xbox Live tant en Windows com en Xbox 360 i una millor gestió dels dispositius.
 - XNA Game Studio 3.0: Suporta Visual Studio 2008, C# 3.0, la possibilitat de crear versions de prova dels jocs, diverses característiques multijugador a través de Xbox Live i la possibilitat de crear jocs multiplataforma que funcionin en Windows, Xbox 360 i Zune.
 - XNA Game Studio 3.1: Inclou la possibilitat de reproduir vídeos, una millor API d'àudio, un sistema per a poder gestionar grups de persones connectades a Xbox Live i permet la utilització dels avatars de la Xbox 360 en els jocs.
 - XNA Game Studio 4.0: Afegeix el suport per a la plataforma Windows Phone 7, perfils de hardware, efectes configurables i integració de Visual Studio 2010, entre d'altres.
- XNA Framework Content Pipeline: És un conjunt d'eines que permet al programa Visual Studio i XNA Studio actuar com a punt de disseny clau per a organitzar el contingut 3D.

Per treballar amb XNA n'hi ha prou en tenir alguna de les versions de Visual Studio, que ja permetria poder desenvolupar jocs per a Windows del tipus Windows Game. Si el que es vol és desenvolupar per a Xbox 360, els jocs es compilen directament a la consola. Per això, la Xbox i l'ordinador en el que es desenvolupa el joc han d'estar comunicats, ja sigui mitjançant un cable de xarxa o l'antena Wi-Fi.

Un cop comunicats, s'executa a la Xbox el programa XNA Game Studio Connect, que queda a l'espera que l'ordinador li envii el codi que ha de compilar. Si el Visual Studio està configurat per compilar el joc a la Xbox, cada cop que es compili el codi s'enviarà una còpia del joc a la consola, i al cap d'uns instants es podran veure els resultats a la pantalla.

Quan es comença un projecte de Xbox 360 nou en XNA 4.0, Visual Studio crea dos apartats pel projecte: la part de codi i la part de contingut (models, imatges, sons...). Per la part de codi, Visual Studio afegeix les propietats i referències del projecte, icones del joc, la classe `Game1.cs` (si al crear el projecte no s'ha modificat el seu nom) on es troba el codi principal del joc i la classe `Program.cs`, que conté el llançador del projecte.

També permet incorporar a la classe `Game1.cs` l'estructura bàsica de codi per a desenvolupar un joc. Aquesta plantilla està dividida en varis apartats:

- Les llibreries que utilitza la classe.
- El nom del projecte al que pertany la classe.
- La classe pública `Game1`. Aquí s'inclou el gruix del codi i es troben els següents subapartats:
 - o Declaracions de variables globals.
 - o Constructor `Game1()`: Inicialitza una nova instància de la classe, que proporciona una inicialització bàsica del dispositiu de gràfics, la lògica del joc, el codi de renderització i un bucle de joc.
 - o Mètode `Initialize()`: Aquí es poden inicialitzar els recursos que no necessiten la classe `GraphicsDevice` per fer-ho.
 - o Mètode `LoadContent()`: Aquí es pot carregar qualsevol recurs que utilitzi el joc, com un model o una textura.
 - o Mètode `UnloadContent()`: Aquí és on s'alliberen els recursos. Com que en principi quan un recurs ja no s'utilitza s'allibera automàticament, aquest mètode acostuma a estar buit.
 - o Mètode `Update()`: Aquest és un mètode cíclic que no para de cridar-se cada vegada. Aquí és on s'actualitza la lògica del joc.
 - o Mètode `Draw()`: Aquest mètode també és cíclic, i és on es renderitzen tots els aspectes visuals del joc a la pantalla.

3.3 XACT

XNA Framework ofereix reproducció d'àudio a través de l'eina Microsoft Cross-Platform Audio Creation Tool (XACT) i una classe `SoundEffect` senzilla que no necessita XACT. XACT només és compatible amb les plataformes Windows i Xbox 360.

El programa XACT Audio Authoring Tool és una aplicació inclosa dins del XNA Game Studio que permet organitzar arxius d'àudio en *wave banks* (bancs d'arxius que inclouen arxius WAV) i *sound banks* (col·leccions que contenen les instruccions de reproducció dels *wave banks*). Tots aquests sons es reproduïxen

mitjançant *cues*. Aquestes indicacions es componen d'un o més sons i es fan referència a través d'un *sound bank*.

Al ser un sistema de creació de contingut d'àudio controlat per una interfície d'usuari gràfica, XACT permet als dissenyadors d'àudio carregar els arxius wave en grups, organitzar els arxius en indicacions discretes que puguin activar els esdeveniments i crear transicions entre indicacions. XACT també permet definir variables que es puguin canviar en el joc per a modificar la configuració d'àudio.

3.4 C#

Degut a les limitacions a l'hora de programar per a Xbox 360, el projecte s'ha de programar en C#.

C# és un llenguatge de programació orientat a objectes desenvolupat i estandarditzat per Microsoft com a part de la seva plataforma .NET. Actualment la versió estable més recent és C# 4.0, tot i que ja s'està preparant la versió 5.0. Els objectius d'aquest llenguatge són:

- Ser simple, modern, de propòsit general i orientat a objectes.
- Aquest llenguatge i les seves implementacions han de complir amb els principis de l'enginyeria de software, tals com la comprovació de tipus, comprovació de límits en matrius, detecció d'ús de variables no inicialitzades i el recollidor automàtic de porqueria.
- El llenguatge està pensat per utilitzar-se en la programació de components en entorns distribuïts.
- Reaprofitar tant com sigui possible els coneixements dels llenguatges de programació C i C++.

3.5 Visual Studio

Visual Studio és un entorn integrat de desenvolupament creat per Microsoft que permet desenvolupar aplicacions tant de consola com d'interfície gràfica, entre d'altres. El programa inclou un editor de codi, un depurador integrat, un dissenyador d'interfícies gràfiques, un dissenyador web, un dissenyador de classes i un dissenyador d'esquemes de bases de dades.

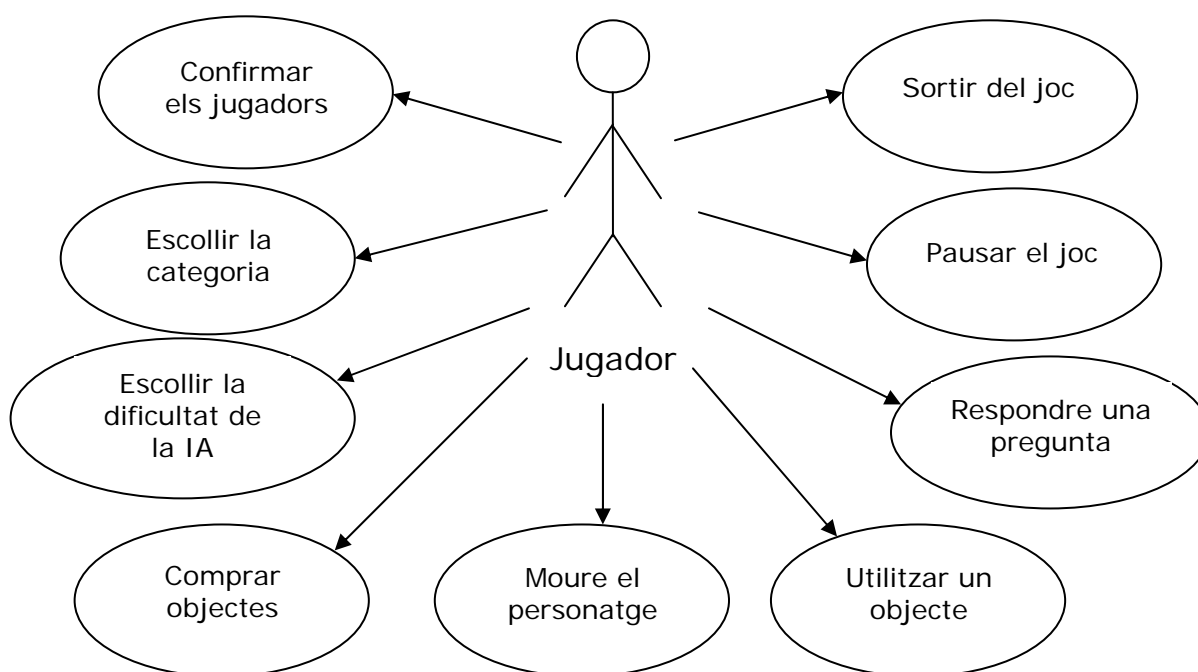
Suporta varis llenguatges de programació de forma nativa, com C/C++, VB.NET, C#, XML/XSLT, HTML/XHTML, JavaScript i CSS. A més també és possible afegir-hi altres llenguatges per separat, com M, Python i Ruby.

Existeixen diverses edicions del programa amb diferents característiques. La versió estable més nova és la Visual Studio 2010, tot i que la Visual Studio 11 ja està en fase beta. Les edicions més bàsiques són gratuïtes, i a mesura que es van afegint característiques el preu també s'eleva. També hi ha versions que suporten únicament un tipus de llenguatges concrets, com Visual C#.

4. Anàlisis i disseny del joc

Com ja s'ha explicat anteriorment, existeixen molts tipus i gèneres de videojocs. Com que no es pot encaixar aquest projecte exactament en algun d'aquests gèneres, en aquest apartat es farà un repàs general del joc i es mostrarà pas a pas tot el que ofereix.

4.1 Diagrama dels casos d'ús



4.2 Especificació dels casos d'ús

Confirmar quants jugadors jugaran la partida	
Descripció	Abans de començar la partida, els jugadors han de confirmar la seva intenció de participar-hi.
Pre-condicions	<ol style="list-style-type: none"> 1. El joc ha d'estar a la pantalla del menú principal. 2. No pot haver un nombre superior a 4 jugadors amb intenció de jugar la mateixa partida.
Flux bàsic	<ol style="list-style-type: none"> 1. Un jugador prem el botó <START> del seu comandament. 2. Apareix la pantalla de selecció de jugadors. 3. Si un jugador té el seu gamertag enregistrat a la consola, pot iniciar sessió amb ell. 4. Quan tots els jugadors estan llestos, qualsevol

	jugador prem el botó <START> per continuar.
Flux alternatiu 1: Un jugador no té un gamertag a la consola	<ol style="list-style-type: none"> 1. Un jugador prem el botó <START> del seu comandament. 2. Apareix la pantalla de selecció de jugadors. 3. Si un jugador no té el seu gamertag enregistrat a la consola, ha de crear-se'n un mitjançant les opcions del sistema operatiu de la Xbox. 4. Un cop creat, ja pot iniciar sessió. 5. Quan tots els jugadors estan llestos, qualsevol jugador prem el botó <START> per continuar.
Post-condicions	<ol style="list-style-type: none"> 1. La configuració de jugadors s'ha de mantenir durant tota la partida fins que es surti del joc. 2. Si no s'han afegit 4 jugadors, els llocs lliures que sobrin fins arribar a 4 seran completats amb personatges controlats per la IA, amb aparença aleatòria.

Escollir la dificultat dels personatges controlats per la IA	
Descripció	Els personatges controlats per la IA tenen varis nivells de dificultat, de forma que els jugadors poden ajustar-lo al que més els hi convingui per fer el joc més fàcil o més difícil.
Pre-condicions	<ol style="list-style-type: none"> 1. El joc ha de venir de la pantalla de títol, o bé del menú de pausa durant la partida. 2. Els jugadors reals que s'han afegit a la partida han de ser com a molt tres, per així afegir com a mínim un personatge controlat per la IA.
Flux bàsic	<ol style="list-style-type: none"> 1. En la pantalla del menú principal es mostra un llistat amb el nombre de jugadors que han iniciat la sessió. Els que faltin fins arribar a quatre seran els que controla la IA. Cada personatge controlat per la IA té el nivell de dificultat "Normal" ajustat per defecte. 2. Si els jugadors volen canviar el nivell de dificultat d'alguns dels personatges controlats per la màquina, han de seleccionar l'opció del menú "Selecciona dificultat IA". 3. Un cop al selector de dificultat, qualsevol jugador pot prémer el botó <A> del comandament per canviar el nivell de dificultat, podent escollir entre "Fàcil", "Normal" i "Difícil".

	<ol style="list-style-type: none"> 4. Quan el nivell de dificultat estigui ajustat correctament, qualsevol jugador pot prémer el botó <START> per continuar.
Flux alternatiu 1: Es vol canviar el nivell de la IA durant la partida.	<ol style="list-style-type: none"> 1. Mentre els jugadors estan a mitja partida, qualsevol d'ells prem el botó <START> per pausar el joc. 2. Un cop pausat el joc, el jugador ha de seleccionar "Opcions" i prémer el botó <A>. 3. Apareix el submenú d'opcions. Ara el jugador ha de seleccionar "Canvia la dificultat de la IA" i prémer el botó <A>. Això farà canviar el nivell de dificultat. Aquest pas s'ha de repetir fins a trobar el nivell de dificultat desitjat. 4. Un cop trobat, el jugador ha de seleccionar l'opció "Tornar" i prémer <A>. Llavors, en el menú de pausa, seleccionar "Continuar" i tornar a prémer <A>.
Post-condicions	<ol style="list-style-type: none"> 1. La dificultat dels personatges controlats per la IA s'ha de mantenir durant tota la partida fins que es surti del joc, es torni a la pantalla del menú principal o es modifiqui mitjançant el menú de pausa.

Escollir la categoria de les preguntes	
Descripció	El joc conté diverses categories de preguntes i cada categoria conté preguntes diferents. Els jugadors poden escollir de quina categoria volen que se'ls preguntin mitjançant aquesta opció.
Pre-condicions	<ol style="list-style-type: none"> 1. El joc ha de venir de la pantalla de títol, o bé del menú de pausa durant la partida.
Flux bàsic	<ol style="list-style-type: none"> 1. En la pantalla del menú principal es mostra la categoria amb la que es jugarà la partida. Per defecte, la categoria seleccionada és "Categoria 1". 2. Si es vol canviar, qualsevol jugador ha de seleccionar l'opció "Selecciona categoria" i prémer el botó <A> perquè el joc mostri la següent categoria disponible. Aquest pas s'ha de repetir tants cops com sigui necessari fins a trobar la categoria desitjada. 3. Quan es trobi la categoria que es vol jugar, qualsevol jugador pot prémer el botó <START> per continuar.

<p>Flux alternatiu 1: Es vol canviar la categoria de les preguntes durant la partida.</p>	<ol style="list-style-type: none"> 1. Mentre els jugadors estan a mitja partida, qualsevol d'ells prem el botó <START> per pausar el joc. 2. Un cop pausat el joc, el jugador ha de seleccionar "Opcions" i prémer el botó <A>. 3. Apareix el submenú d'opcions. Ara el jugador ha de seleccionar "Categoria de les preguntes" i prémer el botó <A>. Això farà canviar la categoria. Aquest pas s'ha de repetir fins a trobar la categoria desitjada. 4. Un cop trobada, el jugador ha de seleccionar l'opció "Tornar" i prémer <A>. Llavors, en el menú de pausa, seleccionar "Continuar" i tornar a prémer <A>.
<p>Post-condicions</p>	<ol style="list-style-type: none"> 1. La categoria de les preguntes s'ha de mantenir durant tota la partida fins que es surti del joc, es torni a la pantalla del menú principal o es modifiqui mitjançant el menú de pausa.

<p>Comprar objectes a la tenda</p>	
<p>Descripció</p>	<p>La tenda permet comprar objectes d'atac i defensa que es poden utilitzar durant la partida, donant avantatge al jugador que els utilitza.</p>
<p>Pre-condicions</p>	<ol style="list-style-type: none"> 1. El joc ha de venir de la pantalla del menú principal o de la tercera ronda d'una partida en curs. 2. Tots els jugadors de la partida han d'estar configurats. 3. La categoria de les preguntes de la partida ha d'haver estat escollida.
<p>Flux bàsic</p>	<ol style="list-style-type: none"> 1. Es mostren varis objectes d'atac i de defensa, on cada objecte té un preu. 2. El jugador posa el cursor sobre l'objecte que vol comprar i prem el botó <A> per confirmar la compra. 3. El preu de l'objecte es resta dels punts acumulats que té el jugador. 4. Si el jugador vol comprar un altre objecte torna al punt 2 del flux. 5. Si el jugador ja no vol o no pot comprar més objectes, prem el botó <START> per continuar.

<p>Flux alternatiu 1: Hi ha més d'una persona jugant</p>	<ol style="list-style-type: none"> 1. Es mostren varis objectes d'atac i de defensa, on cada objecte té un preu. 2. El jugador posa el cursor sobre l'objecte que vol comprar i prem el botó <A> per confirmar la compra. 3. El preu de l'objecte es resta dels punts acumulats que té el jugador. 4. Si el jugador vol comprar un altre objecte torna al punt 2 del flux. 5. Si el jugador ja no vol o no pot comprar més objectes, prem el botó <START> per continuar. 6. Es torna al punt 2 del flux, però aquest cop el control del cursor el té el següent jugador disponible, repetint la seqüència tants cops com persones reals s'hagin afegit a la partida.
<p>Flux alternatiu 2: El jugador no té suficients punts per comprar un objecte</p>	<ol style="list-style-type: none"> 1. Un jugador selecciona l'objecte que vol comprar però no té suficients punts per fer-ho, i prem el botó <A> del comandament. 2. El joc indica al jugador que no té suficients punts per comprar l'objecte i se li cancel·la la compra. 3. El jugador pot intentar comprar un altre objecte que costi menys punts, o pot prémer el botó <START> per continuar.
<p>Post-condicions</p>	<ol style="list-style-type: none"> 1. Cada objecte es pot utilitzar tantes vegades com objectes del mateix tipus hagi comprat el jugador. 2. Els objectes que hagin quedat sense utilitzar al final d'una partida, es mantenen a la següent. 3. Els objectes es perden si es torna a la pantalla de títol o es surt del joc.

<p>Moure el personatge</p>	
<p>Descripció</p>	<p>Un cop ha començat la partida, els personatges s'han de moure per l'escenari per poder arribar a la zona de respostes i poder contestar.</p>
<p>Pre-condicions</p>	<ol style="list-style-type: none"> 1. La partida ha d'haver començat.
<p>Flux bàsic</p>	<ol style="list-style-type: none"> 1. El jugador mou el joystick esquerre del comandament cap a una direcció. 2. El joc analitza la direcció que el jugador està enviant a la consola.

	<ol style="list-style-type: none"> 3. El joc modifica l'animació del personatge, que passa a "caminar" o "córrer" segons el grau d'inclinació del joystick. 4. El joc modifica la posició del personatge segons la direcció en la que s'inclina el joystick.
Flux alternatiu 1: El personatge topa contra un obstacle	<ol style="list-style-type: none"> 1. El jugador mou el joystick esquerre del comandament cap a una direcció. 2. El joc analitza la direcció que el jugador està enviant a la consola. 3. El joc modifica l'animació del personatge, que passa a "caminar" o "córrer" segons el grau d'inclinació del joystick. 4. El joc modifica la posició del personatge segons la direcció en la que s'inclina el joystick. 5. El personatge topa contra un obstacle (objecte, paret). 6. El joc continua mostrant l'animació "caminar" o "córrer" pel personatge, però no modifica la seva posició. 7. El jugador ha de modificar la direcció en la que fa moure el personatge per continuar endavant.
Post-condicions	<ol style="list-style-type: none"> 1. El personatge es continuarà movent cap a la direcció que vulgui el jugador fins que aquest deixi el joystick en una posició neutre.

Utilitzar un objecte	
Descripció	Durant la partida, els jugadors poden utilitzar objectes per atacar-se entre ells o utilitzar-los a sí mateixos per obtenir algun tipus de millora envers els demés.
Pre-condicions	<ol style="list-style-type: none"> 1. La partida ha d'estar en marxa i no pot estar pausada. 2. El jugador ha de tenir com a mínim un objecte abans de començar la partida.
Flux bàsic	<ol style="list-style-type: none"> 1. El jugador prem el botó <A> per utilitzar l'atac 1. 2. El joc comprova la posició del jugador i calcula quins són els contrincants que estan dins del radi d'acció. 3. Un cop calculats, la barra d'energia d'aquests

	jugadors es redueix un 50%.
Flux alternatiu 1: El personatge utilitza l'atac 2	<ol style="list-style-type: none"> 1. El jugador prem el botó per utilitzar l'atac 2. 2. La barra d'energia de tots els contrincants es redueix un 40%, sense importar la distància a la que es trobin.
Flux alternatiu 2: El personatge utilitza la defensa 1	<ol style="list-style-type: none"> 1. El jugador prem el botó <X> per utilitzar la defensa 1. 2. La barra d'energia del jugador augmenta 50 punts.
Flux alternatiu 3: El personatge utilitza la defensa 2	<ol style="list-style-type: none"> 1. El jugador prem el botó <Y> per utilitzar la defensa 2. 2. El joc va seguint la posició del personatge i genera un escut envers ell durant uns segons.
Post-condicions	<ol style="list-style-type: none"> 1. El jugador que hagi utilitzat un objecte se li resta una unitat del tipus que hagi fet servir. 2. Si un jugador resulta atacat durant la partida, guanya punts de defensa.

Respondre una pregunta	
Descripció	L'objectiu de la partida és respondre correctament les preguntes, de forma que es pugui guanyar més punts d'atac que els demés jugadors.
Pre-condicions	<ol style="list-style-type: none"> 1. La partida ha d'estar en marxa i no pot estar pausada.
Flux bàsic	<ol style="list-style-type: none"> 1. El jugador arriba a la zona de respostes i col·loca el seu personatge dins de la zona que correspon a una resposta concreta. 2. El joc detecta que el jugador ha entrat dins d'una zona de resposta i comença un compte enrere de 5 segons. 3. Quan s'acaba el compte enrere, la resposta que ha escollit el jugador i la posició en la que ha respòs la pregunta queden registrades pel joc i el personatge queda immobilitzat.
Flux alternatiu 1: El personatge surt de la zona de respostes	<ol style="list-style-type: none"> 1. El jugador arriba a la zona de respostes i col·loca el seu personatge dins de la zona que correspon a una resposta concreta. 2. El joc detecta que el jugador ha entrat dins d'una zona de resposta i comença un compte enrere de 5 segons. 3. El jugador surt de la zona de respostes, ja sigui per iniciativa pròpia o perquè altres

	<p>jugadors el fan fora abans que s'acabin els 5 segons.</p> <p>4. El compte enrere es reinicia i el jugador ha de tornar a entrar dins d'una zona de resposta.</p>
Post-condicions	<p>1. Si el jugador ha respòs correctament la pregunta i és el primer en fer-ho, guanya 30 punts d'atac.</p> <p>2. Si el jugador ha respòs correctament la pregunta i és el segon en fer-ho, guanya 20 punts d'atac.</p> <p>3. Si el jugador ha respòs correctament la pregunta i és el tercer en fer-ho, guanya 10 punts d'atac.</p> <p>4. Si el jugador ha respòs correctament la pregunta i és el quart en fer-ho, guanya 5 punts d'atac.</p> <p>5. Si el jugador ha respòs incorrectament la pregunta, no suma cap punt d'atac.</p>

Pausar el joc	
Descripció	Durant la partida, interrompre el joc serveix tant per aturar de forma momentània el joc com per accedir a les opcions del menú de pausa.
Pre-condicions	<p>1. La partida ha d'estar en marxa i no pot estar pausada.</p>
Flux bàsic	<p>1. La partida està en marxa i qualsevol jugador prem el botó <START>.</p> <p>2. El joc paralitza la partida i mostra un menú amb vèries opcions.</p> <p>3. Per tornar a posar en marxa el joc, el jugador que l'ha interromput prem de nou el botó el botó <START> o bé selecciona l'opció "Continuar" i prem el botó <A>.</p> <p>4. El joc continua de la mateixa forma que estava anteriorment.</p>
Flux alternatiu 1: Pausar el joc i seleccionar un canvi de categoria de preguntes	<p>1. La partida està en marxa i qualsevol jugador prem el botó <START>.</p> <p>2. El jugador que ha interromput el joc situa el cursor sobre l'opció "Categoria de les preguntes" i prem el botó <A>.</p> <p>3. La categoria de les preguntes canvia. S'ha de repetir el pas 3 fins a trobar la categoria que es vulgui.</p>

Flux alternatiu 2: Pausar el joc i seleccionar tornar al títol del joc	<ol style="list-style-type: none"> 1. La partida està en marxa i qualsevol jugador prem el botó <START>. 2. El jugador que ha interromput el joc situa el cursor sobre l'opció "Tornar al títol" i prem el botó <A>. 3. El joc elimina les variables temporals de la partida en marxa i torna a la pantalla de títol.
Flux alternatiu 3: Pausar el joc i seleccionar sortir del joc	<ol style="list-style-type: none"> 1. La partida està en marxa i qualsevol jugador prem el botó <START>. 2. El jugador que ha interromput el joc situa el cursor sobre l'opció "Sortir del joc" i prem el botó <A>. 3. El joc allibera la memòria del programa i torna al menú principal de la consola.

Sortir del joc	
Descripció	Els jugadors volen acabar el joc i tornar al menú principal de la consola.
Pre-condicions	<ol style="list-style-type: none"> 1. Si es vol sortir del joc a través del menú de pausa, la partida ha d'estar en marxa però en estat de pausa.
Flux bàsic	<ol style="list-style-type: none"> 1. En qualsevol moment del joc, qualsevol jugador prem el botó <BACK>. 2. El joc allibera la memòria del programa i torna al menú principal de la consola
Flux alternatiu 1: Es surt del joc mitjançant el menú de pausa	<ol style="list-style-type: none"> 1. La partida està en marxa i qualsevol jugador prem el botó <START>. 2. El jugador que ha interromput el joc situa el cursor sobre l'opció "Sortir del joc" i prem el botó <A>. 3. El joc allibera la memòria del programa i torna al menú principal de la consola.

4.3 Inici i desenvolupament de la idea

L'objectiu principal del joc és contestar correctament tantes preguntes com sigui possible abans que ho facin els 3 personatges contra els que es competeix. Fins a arribar a aquesta decisió sobre el guió del joc es van barrejar diverses idees.

La idea inicial era desenvolupar amb XNA un joc de minijocs, és a dir, una competició a través de jocs molt simples i curts, i veure quin dels jugadors aconseguia més punts al final. Aquesta idea inicial tenia un parell de problemes. El

principal problema era l'enorme quantitat de feina que portaria pensar i desenvolupar varis minijocs diferents, i més tenint en compte que es partia d'uns coneixements sobre XNA molt bàsics. L'altre problema era la poca originalitat del projecte, ja que no s'intentava innovar en cap aspecte.

És per aquests motius que es va acabar decidint que el millor era centrar-se en un joc i complementar-lo per fer-lo més interessant en comptes de fer-ne varis de curts i senzills. Una de les idees que van sorgir durant el brainstorming va ser la de crear un joc de preguntes i respostes, que es pogués utilitzar, per exemple, per a què els alumnes repassessin els conceptes més importants d'una matèria. A partir d'aquí, es va buscar complementar aquesta idea amb elements que la fessin més atractiva de cara als possibles jugadors i destaqués envers altres tipus de jocs educatius.

Partint de que la base del joc seria la de tenir preguntes i respostes, es va mirar d'accentuar la part competitiva. A partir d'una de les idees que van sorgir per un dels minijocs, es va barrejar el concepte de preguntes i respostes amb el d'una carrera d'obstacles. Els jugadors s'haurien de moure entre els obstacles que hi ha per l'escenari i arribar abans que els altres a la zona de respostes que es troba al final de tot, per poder contestar el primer a la pregunta formulada.

Una altra de les idees per millorar la competició entre els jugadors que es va afegir en el joc va ser la d'utilitzar objectes que servissin per atacar als demés personatges i que quedessin més enrere a la cursa. Ampliant aquesta idea, es va plantejar la inclusió d'objectes de defensa, que contrarestessin l'efecte dels objectes d'atac, o que fins i tot et protegissin d'ells durant uns segons. Això permetria que els jugadors es prenguessin el videojoc com un entreteniment més que com una eina per aprendre.

Pensant en la manera que afectarien els objectes als personatges, es va dur a terme la idea de que cada personatge tingués una energia limitada. L'energia disminueix quan els jugadors corren i es recupera quan es deixen de moure. Si l'energia s'esgota, els personatges es desmaien i no es poden moure durant uns segons. Gràcies a aquest concepte es pot encaixar perfectament la idea dels objectes: els objectes d'atac fan disminuir l'energia dels demés personatges, mentre que els objectes de defensa fan recuperar la seva, o directament protegeixen dels atacs durant un temps limitat.

Amb totes aquestes idees conjuntades el joc agafa molt més cos, mantenint-se fidel a la base de joc amb preguntes i respostes, però jugant amb elements que fan més entretinguda l'experiència i donen més estratègia a la partida.

4.4 Pantalles de títol, opcions i tenda

En els propers apartats s'explica pas a pas tot el que ofereix el joc. A continuació es pot veure el diagrama de dependències entre les classes del projecte.

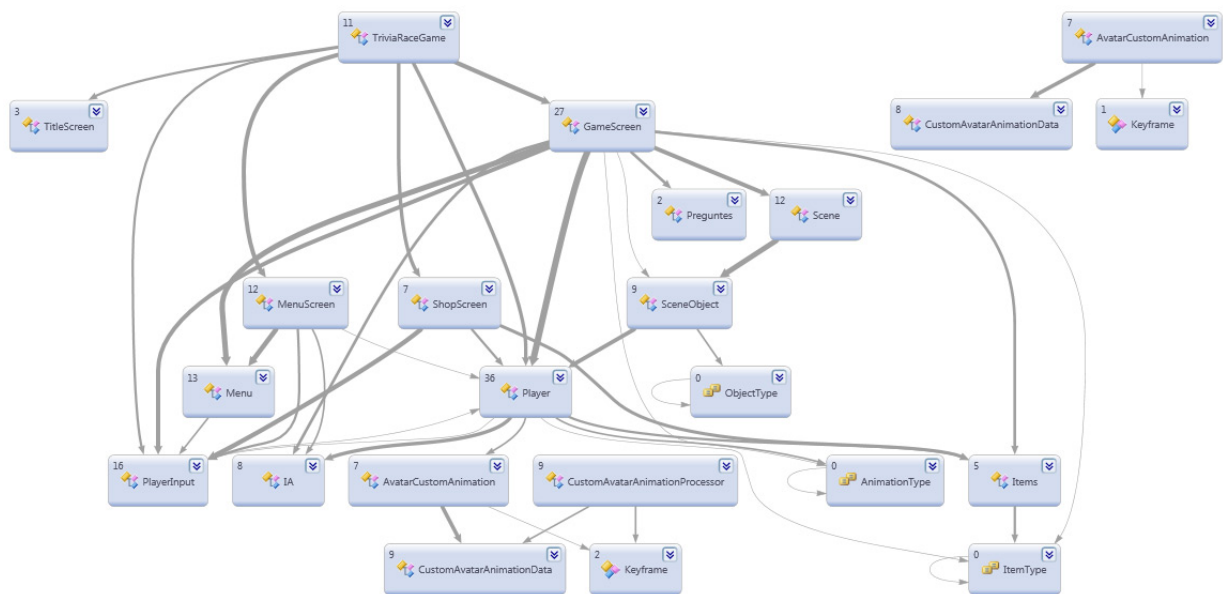


Figura 2: Diagrama de dependències entre classes

Quan s'executa el joc, l'entrada d'execució es realitza a través de la classe TriviaRaceGame.

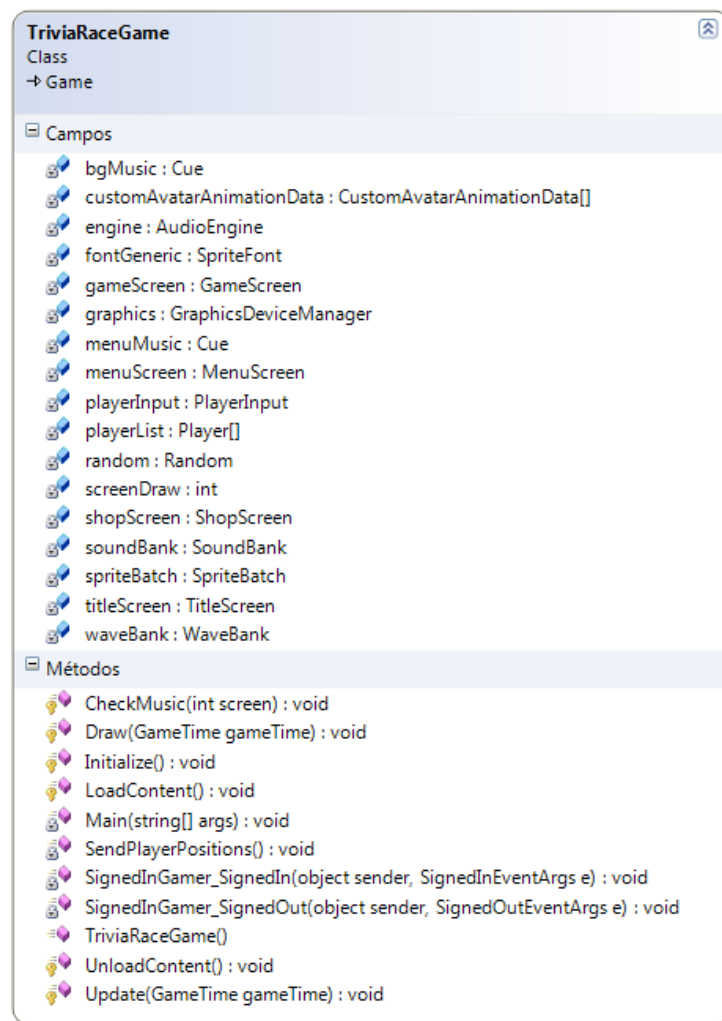


Figura 3: Classe TriviaRaceGame

Aquesta és la classe principal de tot el joc. S'encarrega de crear la llista dels jugadors i les seves variables, controlar quan els jugadors inicien o tanquen la seva sessió a la consola, carregar les animacions personalitzades dels avatars, reproduir la música i els arxius de so i el més important de tot: gestionar les pantalles que es mostraran durant el joc i controlar quina d'elles és la que s'ha de mostrar en cada moment. Sempre està en execució.

L'apartat d'àudio del projecte és molt simple: les pantalles de títol, menú principal i tenda comparteixen una música de fons, mentre que la pantalla de joc té associada una altra pista musical. En quan a efectes sonors, el joc utilitza un so per quan es canvia l'opció ressaltada d'un menú, un altre per quan es modifica o s'accepta una opció del menú, un tercer per quan un personatge utilitza un objecte durant la partida i un últim per quan un personatge és atacat.

Cada jugador utilitza una instància de la classe `Player` per gestionar el seu comportament durant el joc.

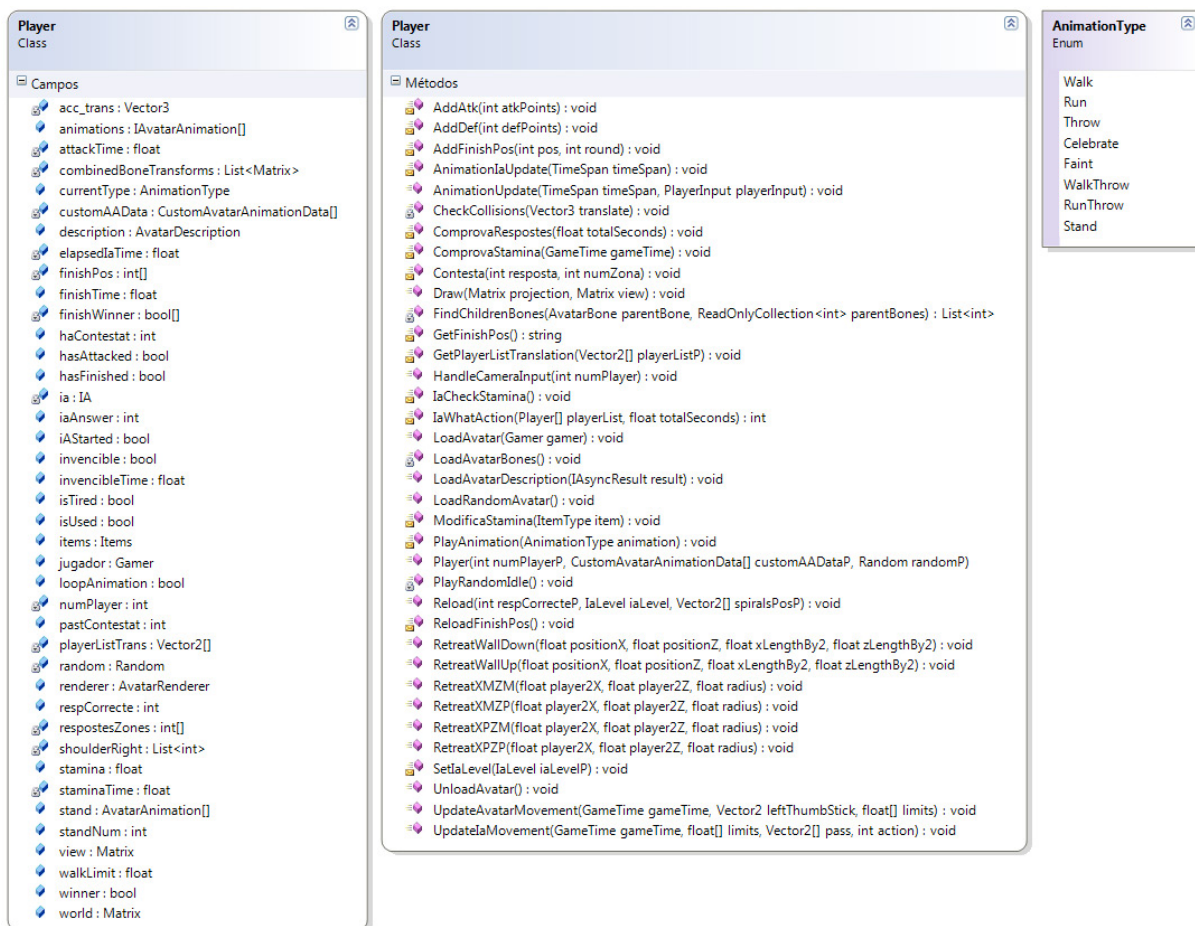


Figura 4: Classe Player

Aquesta classe es va crear per mantenir organitzada tota la informació relacionada amb els personatges. Amb aquesta classe es gestiona tot el que tingui a veure amb els personatges, tant dels jugadors reals com de la IA: el nom i l'aparença que ha de tenir l'avatar del jugador, decidir quina animació s'ha de mostrar en cada moment i mostrar-la correctament, el punt de l'escenari en el que es troba l'avatar i el seu moviment per l'escenari, comprovar les col·lisions contra els objectes de l'escenari i els altres personatges, si ha contestat o no la resposta, en quina posició ha contestat la resposta, si ha guanyat la ronda de joc, gestionar l'energia del jugador segons hagi d'augmentar o disminuir i si està cansat perquè se li ha esgotat l'energia.



Figura 5: Avatars

La classe també controla si el personatge el porta un jugador real o la IA. Si el personatge està controlat per la IA, també es gestiona com ha d'actuar en cada moment segons el nivell que se li ha assignat. La decisió de la IA de prendre el camí més curt o de contestar correctament a la pregunta no es realitza directament a la classe Player, si no mitjançant la classe IA.

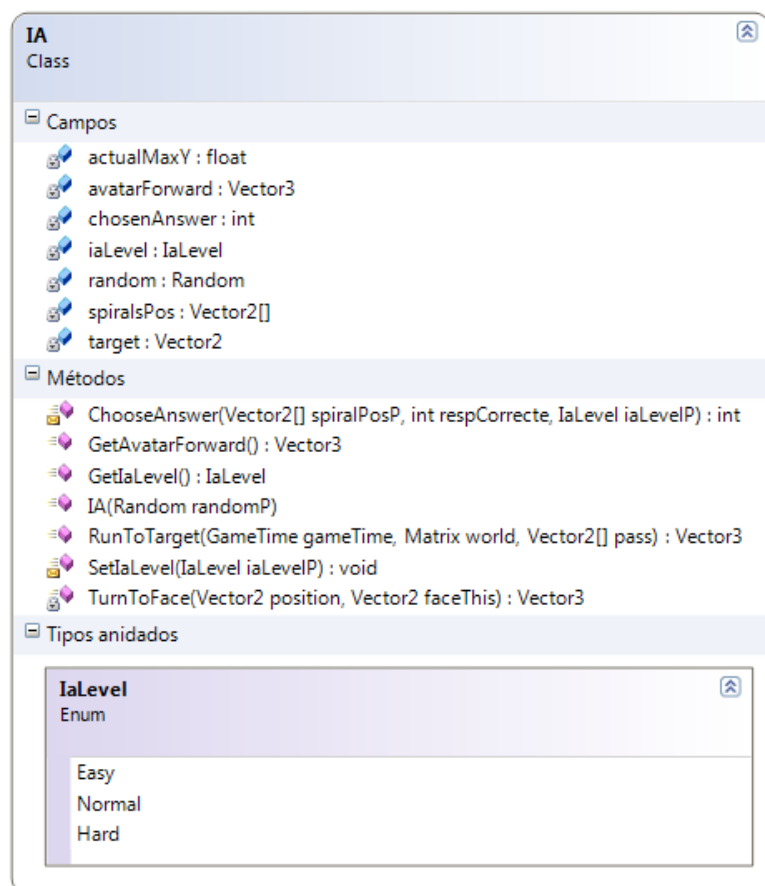


Figura 6: Classe IA

A la classe IA, el mètode RunToTarget() decideix quin és el camí que ha de seguir el personatge controlat per la IA i el mètode ChooseAnswer() decideix la resposta que ha de contestar. Els criteris i regles que segueix la IA per prendre tant aquestes decisions com les de si ha d'utilitzar algun objecte o no s'expliquen més endavant.

En un principi, la classe Player també controlava més aspectes del jugador, però degut a la gran quantitat de variables que ja gestionava, es va decidir separar dos dels seus apartats en dues classes diferents per no saturar-la, tot i que encara en depenen directament: PlayerInput i Items.

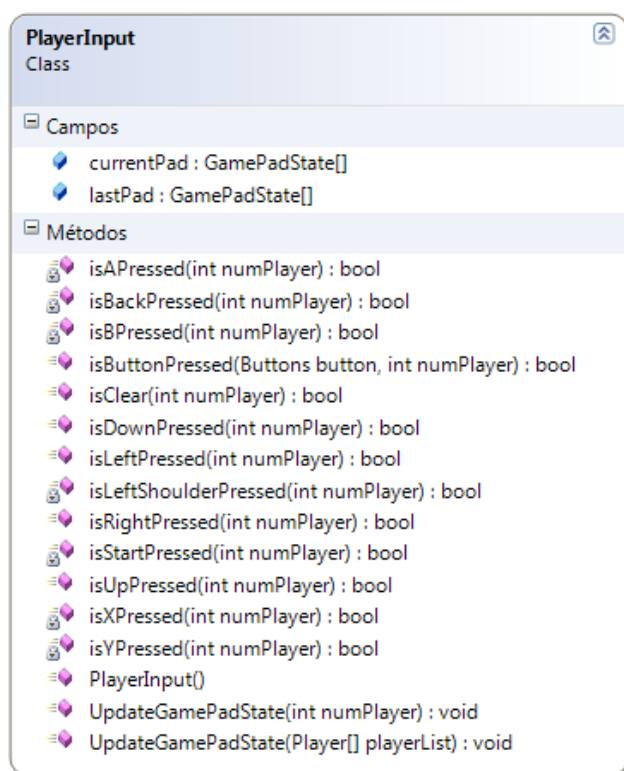


Figura 7: Classe PlayerInput

La classe `PlayerInput` controla en cada moment l'estat del comandament del jugador que té assignat: quins són els botons que està pressionant el jugador o si simplement no n'està pressionant cap. En conseqüència aquesta classe només s'utilitza en el cas que el personatge estigui controlat per un jugador real.

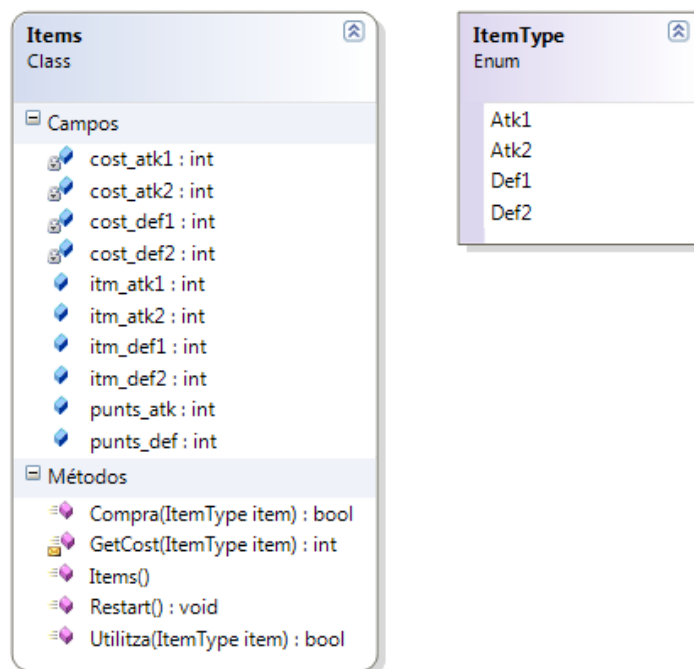


Figura 8: Classe Items

La classe Items gestiona la part que relaciona el jugador amb els objectes que utilitzarà durant la partida i amb el sistema de puntuació. Per una banda, s'encarrega de controlar quina quantitat d'objectes de cada tipus té el jugador, si se n'ha d'afegir perquè n'ha comprat més o si se n'ha de treure perquè n'ha utilitzat algun. Per altra banda, també gestiona els punts d'atac i defensa que té el jugador i el cost que té cada objecte.

L'última part de la que depèn la classe Player és del conjunt de classes i estructures CustomAvatarAnimation. La classe Player s'encarrega de gestionar totes les animacions de l'avatar del jugador, tant de les que ja venen predefinides (com quan el jugador està quiet) com de les personalitzades (caminar, córrer llançar i desmaiar-se). El processador d'animacions per defecte que porta XNA no és capaç d'interpretar les animacions personalitzades que s'hagin afegit externament, i és per això que és necessari crear-ne un de propi.

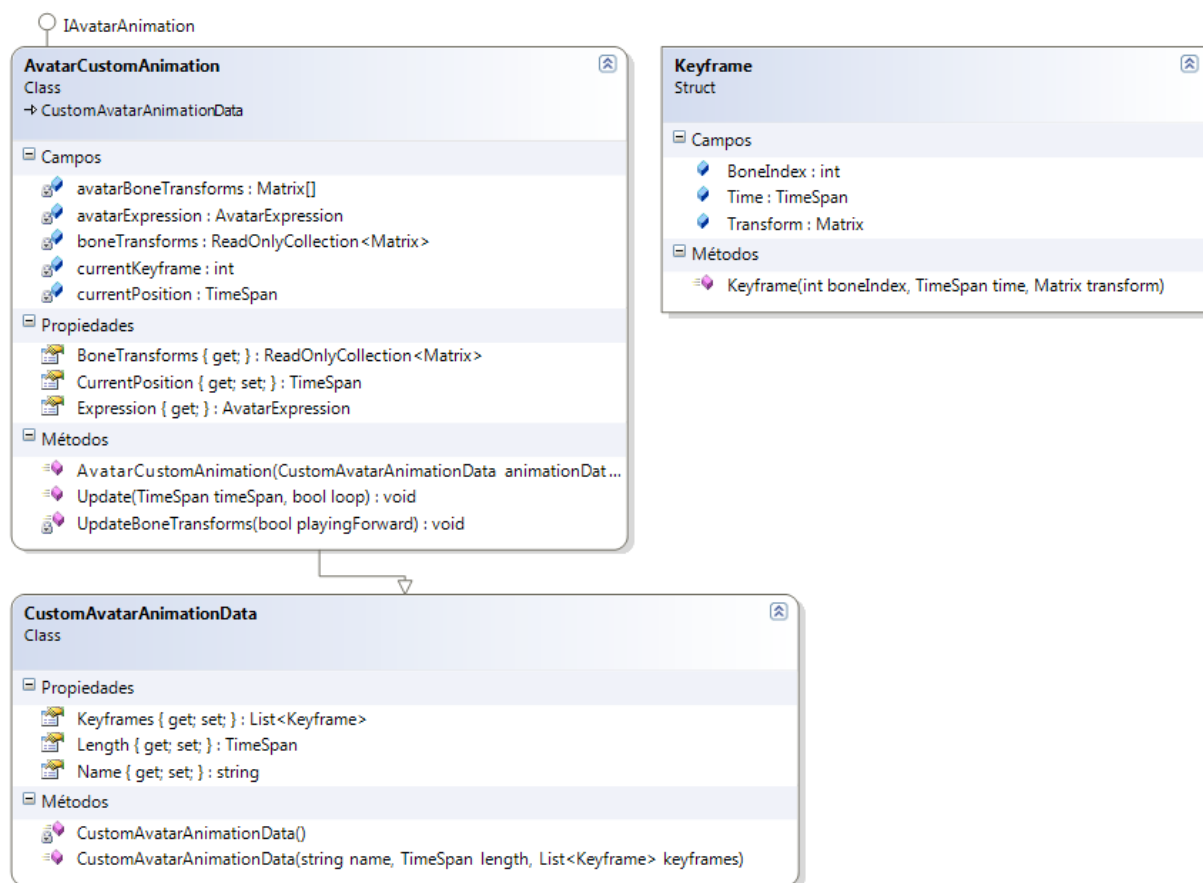


Figura 9: Classes AvatarCustomAnimation i CustomAvatarAnimationData

Aquest processador està compost per les classes AvatarCustomAnimation i CustomAvatarAnimationData, i per l'estructura Keyframe:

- La classe AvatarCustomAnimation s'encarrega de gestionar el moviment dels ossos de l'esquelet de l'avatar, modificant la seva posició a mida que passa el temps. També gestiona què passa un cop s'ha acabat l'animació, si ha de reiniciar-la i tornar a començar o ha de parar-la.
- La classe CustomAvatarAnimationData gestiona el contingut de cada animació personalitzada: els ossos i com estan enganxats cadascun.
- L'estructura Keyframe s'encarrega de relacionar la posició de cada os de l'esquelet de l'avatar amb un moment concret en el temps de l'animació. D'aquesta manera se sap on ha d'anar cada os a cada instant de l'animació.

Les animacions personalitzades que s'han afegit externament al joc (caminar, córrer, llançar un objecte i desmaiar-se) provenen del paquet d'animacions "Avatar Animation Pack" de la secció d'eines per a desenvolupadors de la pàgina App Hub de Microsoft. Aquestes animacions ja venen preparades per a utilitzar-se amb els avatars de la Xbox, facilitant molt la feina a l'hora d'animar els models i podent centrar el temps de desenvolupament més en la part de programació.



Figura 10: Animació personalitzada de desmai

Un cop s'executa el videojoc, apareix la pantalla de títol. Aquesta pantalla és una instància de la classe TitleScreen.

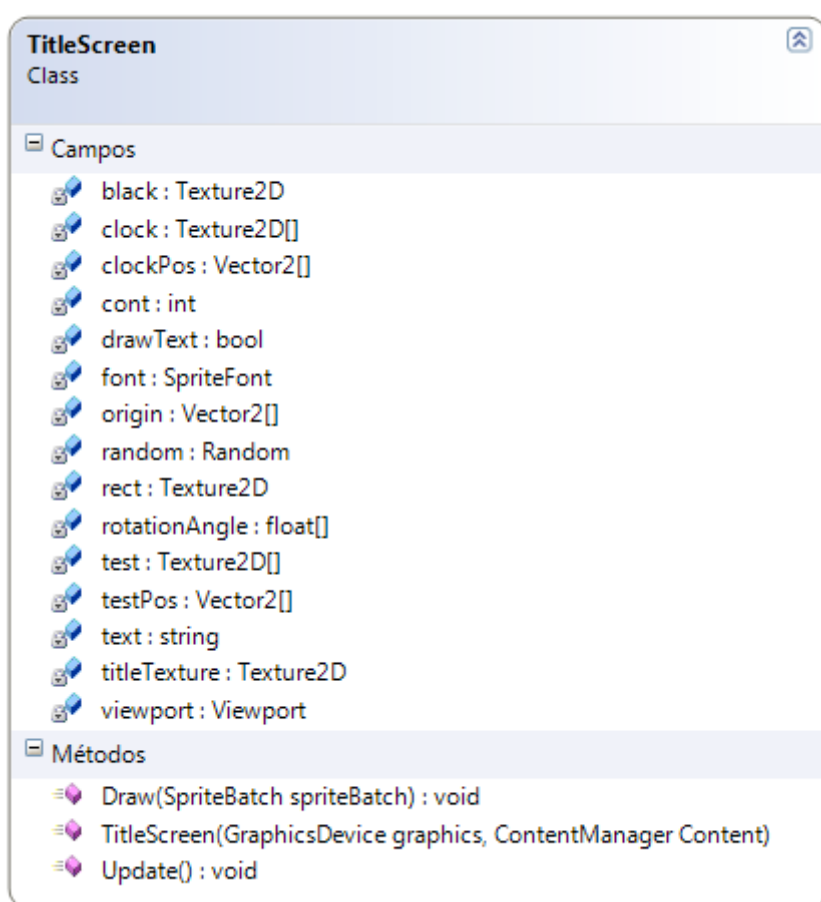


Figura 11: Classe TitleScreen

Aquesta pantalla serveix per donar la benvinguda al joc, i mostra principalment el títol del joc TriviaRace, juntament amb alguns elements gràfics relacionats amb el joc que apareixen i desapareixen per darrere. També es mostra un cartell que indica al jugador que per començar a jugar s'ha de prémer el botó Start del comandament de la Xbox 360. Com que aquesta pantalla només serveix com a

introducció al joc, l'únic que es pot fer aquí és apretar Start i passar a la pantalla del menú principal.



Figura 12: Pantalla TitleScreen

El menú principal el gestiona la classe MenuScreen.



Figura 13: Pantalla MenuScreen

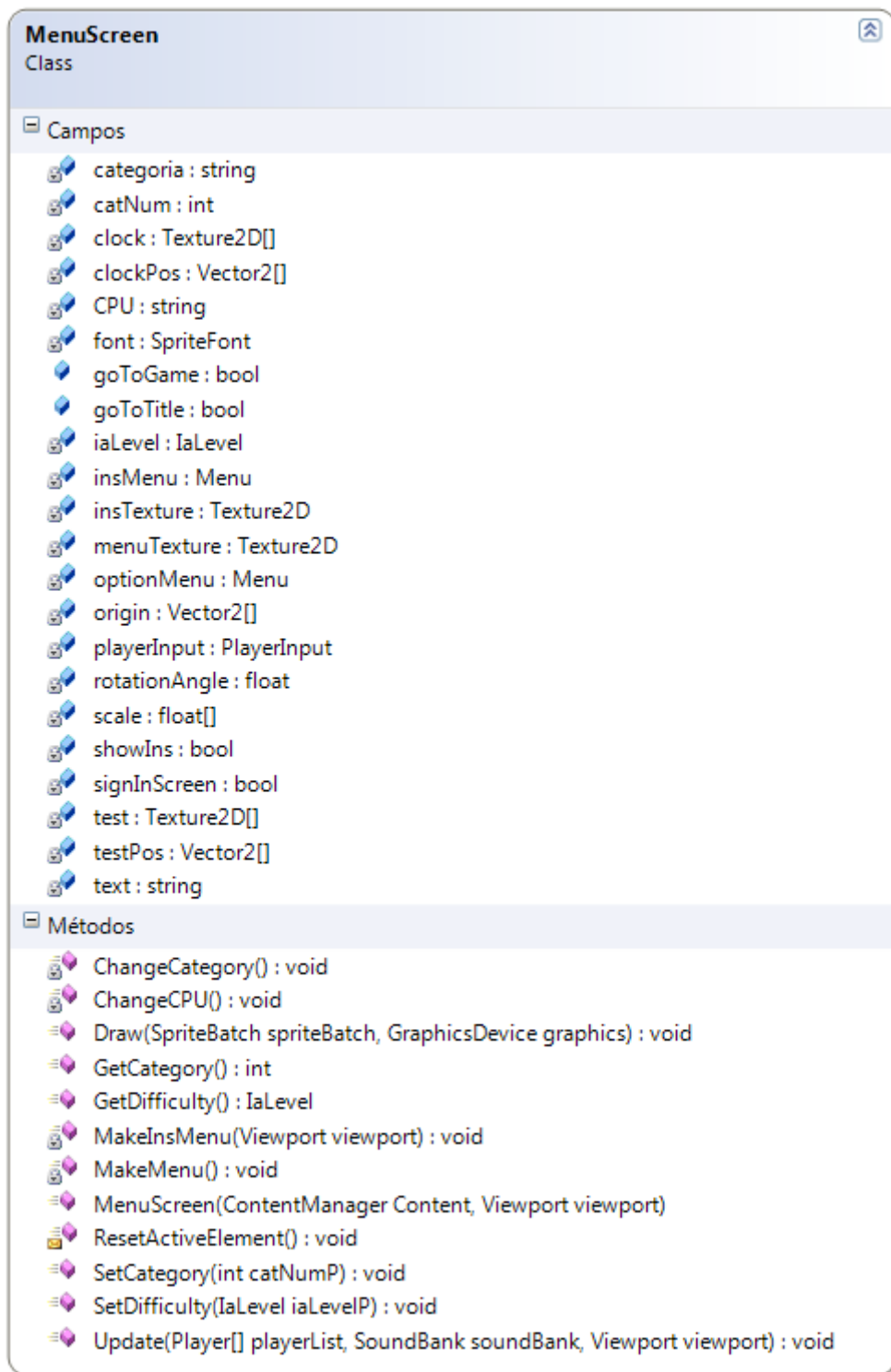


Figura 14: Classe MenuScreen

Aquesta pantalla serveix principalment per configurar la partida i consta de varius apartats. Per una banda, hi ha el logotip del joc juntament amb les opcions del menú principal, mentre que per l'altre hi ha una secció que mostra els jugadors que tenen la seva sessió iniciada. A la part inferior de la pantalla apareixen també els elements gràfics que sortien a la pantalla de títol.

El menú principal, així com altres menús que apareixen en el joc, és una instància de la classe Menu que es veurà més endavant. Per moure's entre les opcions del menú, els jugadors que hagin iniciat sessió poden utilitzar tant el joystick com la creueta del seu comandament Xbox 360. Si premen amunt o avall l'opció del menú ressaltada canvia de posició segons convingui. Per acceptar o modificar l'opció ressaltada (depenent de l'opció) tan sols s'ha de prémer el botó A del comandament.

La primera opció de totes i la principal del menú és "JUGAR". Quan s'accepta l'opció es passa a la pantalla de la tenda per comprar els objectes i començar la partida. La segona opció serveix per modificar la categoria de les preguntes que es realitzaran durant la partida. Hi ha tres categories implementades en el joc: assignatures de la carrera, literatura i cinema. Cada una de les categories té les seves preguntes en un arxiu XML diferent que es llegeix al principi de cada partida.

La tercera opció serveix per modificar el nivell de la IA que tindran els personatges controlats per la consola. Depenent del nivell que es seleccioni, la IA serà més o menys encertada a l'hora de buscar el camí més curt per arribar al final de l'escenari i de contestar la pregunta correctament. La quarta opció mostra les instruccions del joc dins d'un submenú, també creat com a instància de la classe Menu. Quan un jugador vol accedir a les instruccions, la classe MenuScreen deixa de mostrar tots els continguts d'aquesta pantalla i mostra només les instruccions d'ús. A la que es torna del submenú d'instruccions, MenuScreen torna a mostrar el menú principal. Finalment, la última opció serveix per tornar a la pantalla de títol.

A més del menú d'opcions, aquesta pantalla també mostra els jugadors que hi ha identificats. Per jugar, els jugadors necessiten iniciar la sessió al seu compte Xbox. Si un jugador encara no ha iniciat sessió, ha de prémer el botó Start. Al fer-ho, apareix un quadre de diàleg estàndard de Xbox amb tots els comptes que hi ha guardats a la consola perquè el jugador seleccioni mitjançant el botó A del comandament quin compte vol utilitzar. El joc llegeix la informació d'aquest compte i utilitza dos elements: el nom del compte i l'avatar que identifica al jugador. Un cop el jugador ha iniciat la sessió, el nom del seu compte s'afegeix a la llista de jugadors identificats, que es modifica de forma dinàmica.

Com ja s'ha comentat abans, tant el menú principal com el submenú d'instruccions s'han creat mitjançant la classe Menu. Com que el menú d'opcions és un recurs que s'utilitza bastant en el joc, es va decidir separar-lo del codi per poder crear-ne instàncies fàcilment.

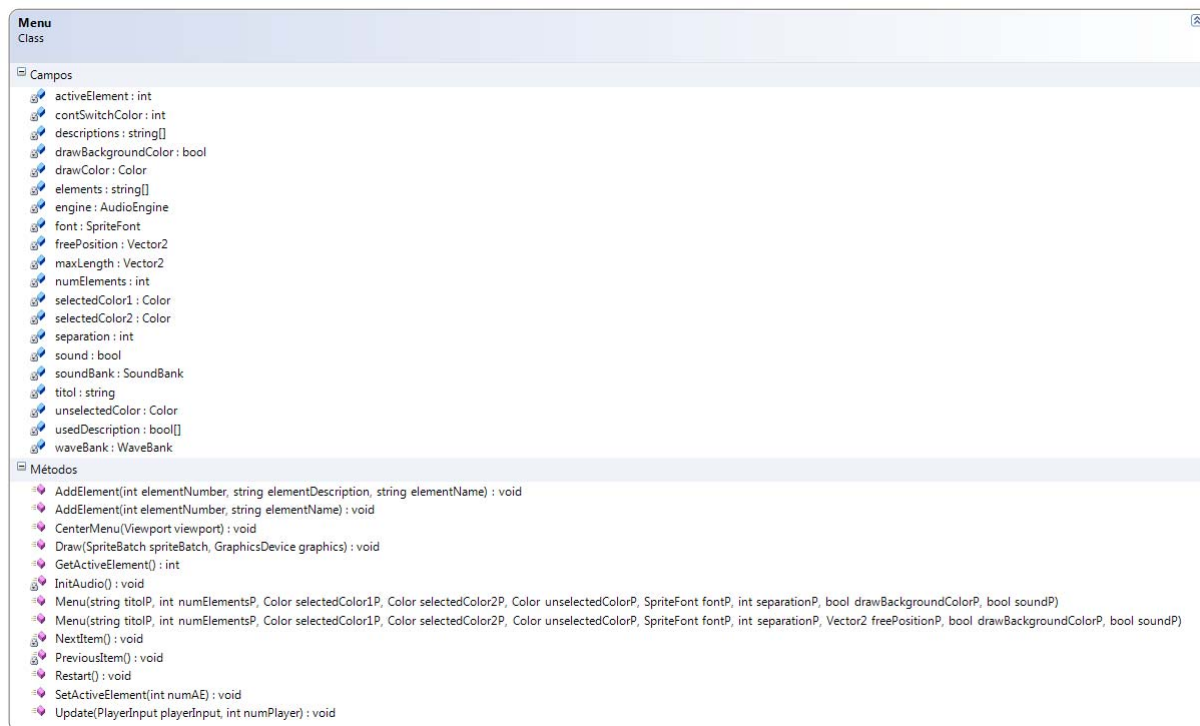


Figura 15: Classe Menu

La classe `Menu` té dos constructors. S'utilitza l'un o l'altre depenent de si es vol que el menú tingui una posició fixa al centre de la pantalla o si es vol posar en un altre punt de forma manual. Amb aquests constructors també es pot decidir el títol del menú, el nombre d'elements que tindrà, els colors que es farà servir pels elements i el fons del menú i si es vol activar o no el so. Aquesta classe també gestiona el moviment de les opcions quan es demana l'element anterior o posterior del menú.

La següent pantalla del joc i la última abans de començar a jugar és la pantalla de tenda. Aquesta pantalla és una instància de la classe `ShopScreen`.

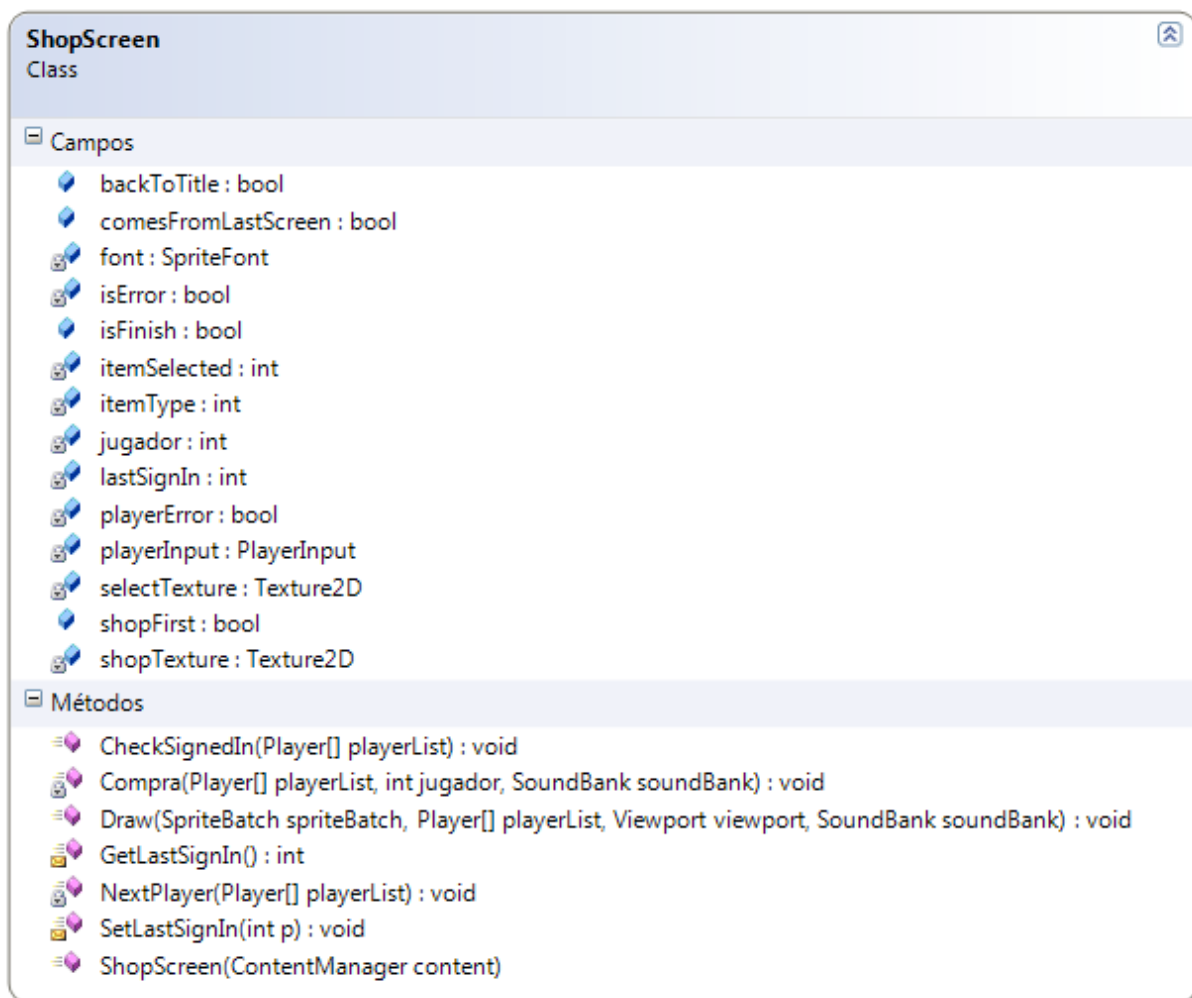


Figura 16: Classe ShopScreen

Aquesta classe s'encarrega de gestionar els torns de compra de tots els jugadors, un per un, fins que no quedi ningú per comprar. Llavors, gestiona la compra que fan els personatges controlats per la IA. Per agilitzar la programació es va decidir no fer distincions segons el nivell de la IA a l'hora de realitzar les seves compres. En tots els casos sempre segueix el mateix ordre de compra fins que el personatge es queda sense punts per comprar res més: Atac 1, defensa 1, atac 1, defensa 1, atac 2 i defensa 2. Això es podria canviar fàcilment, discriminant segons el nivell de la IA, i escrivint decisions més o menys encertades de compra.

En aquesta pantalla es mostren els objectes que hi ha per comprar, la seva descripció, el preu de cada objecte i els punts que té el jugador tant d'atac com de defensa.

A la part superior de la pantalla es mostra quin és el jugador que li toca fer les compres. A la part central/esquerra de la pantalla es mostren en gran els quatre objectes que el jugador pot comprar:

- Atac 1: Està representat per una flama. Serveix per treure el 50% de l'energia dels contrincants que es trobin dins del radi d'acció del jugador que ha utilitzat l'objecte. Costa 5 punts d'atac.
- Atac 2: Està representat per un símbol nuclear. Serveix per treure el 40% de l'energia de tots els contrincants, sense importar la distància amb el jugador que utilitza l'objecte. Costa 10 punts d'atac.
- Defensa 1: Està representat per una creu roja. Serveix perquè el jugador que l'utilitzi recuperi 50 punts d'energia (dels 100 totals). Costa 5 punts de defensa.
- Defensa 2: Està representat per una estrella. Serveix perquè el jugador que l'utilitzi sigui invulnerable als atacs dels seus contrincants durant 10 segons. Costa 10 punts de defensa.

Per seleccionar els diferents objectes, el jugador al que li toqui comprar es pot moure mitjançant el joystick i la creueta del comandament Xbox, prement dreta o esquerra segons convingui. Si decideix comprar algun dels objectes, el jugador ha de prémer el botó A per fer-ho. Si té els suficients punts, es descompta el preu de l'objecte dels punts acumulats i s'afegeix un objecte en el seu inventari. Si no té els punts suficients per comprar-lo, apareix un cartell a la part inferior/dreta de la pantalla que informa que la compra no s'ha dut a terme degut a la falta de punts. A diferència del menú d'opcions de la pantalla anterior, només el jugador al que li toqui comprar podrà moure el cursor.

Cada objecte està acompanyat per un cartell que mostra els punts d'atac o defensa necessaris per comprar-lo, així com de la quantitat actual de cada objecte de la que ja disposa el jugador, segons indiquin les classes Item i Player.

Finalment, a la part central/dreta de la pantalla es mostra per una banda la descripció de l'objecte que està seleccionat en aquell moment, mentre que per l'altre s'indica el nombre de punts restants dels que disposa el jugador, tant d'atac com de defensa. Tots aquests punts es llegeixen de les instàncies de la classe Player. Al començar la partida, tots els jugadors disposen de 50 punts d'atac i 50 punts de defensa per comprar els objectes que considerin més oportuns.

Un cop el jugador ha acabat de comprar tots els objectes que considera necessaris, ha de prémer el botó Start del comandament Xbox per continuar, tal com indica un cartell a la part inferior/esquerra de la pantalla. Si el jugador ho fa i queden més jugadors per comprar, el torn passa al següent jugador de la llista. Si no queden més jugadors, comença la partida.



Figura 17: Pantalla ShopScreen

4.5 Pantalla de joc

La última pantalla del joc és la de la carrera. Aquesta pantalla es controla mitjançant una instància de la classe GameScreen.

Des d'aquesta classe es gestiona la part gràfica i lògica del joc: les divisions que ha de fer la pantalla, l'escenari que s'ha de mostrar, quadrar les posicions i accions dels avatars amb els personatges que es mostren per pantalla, així com controlar que l'input dels comandaments de la Xbox sigui correcte i si s'ha d'activar la pausa en el joc o no.

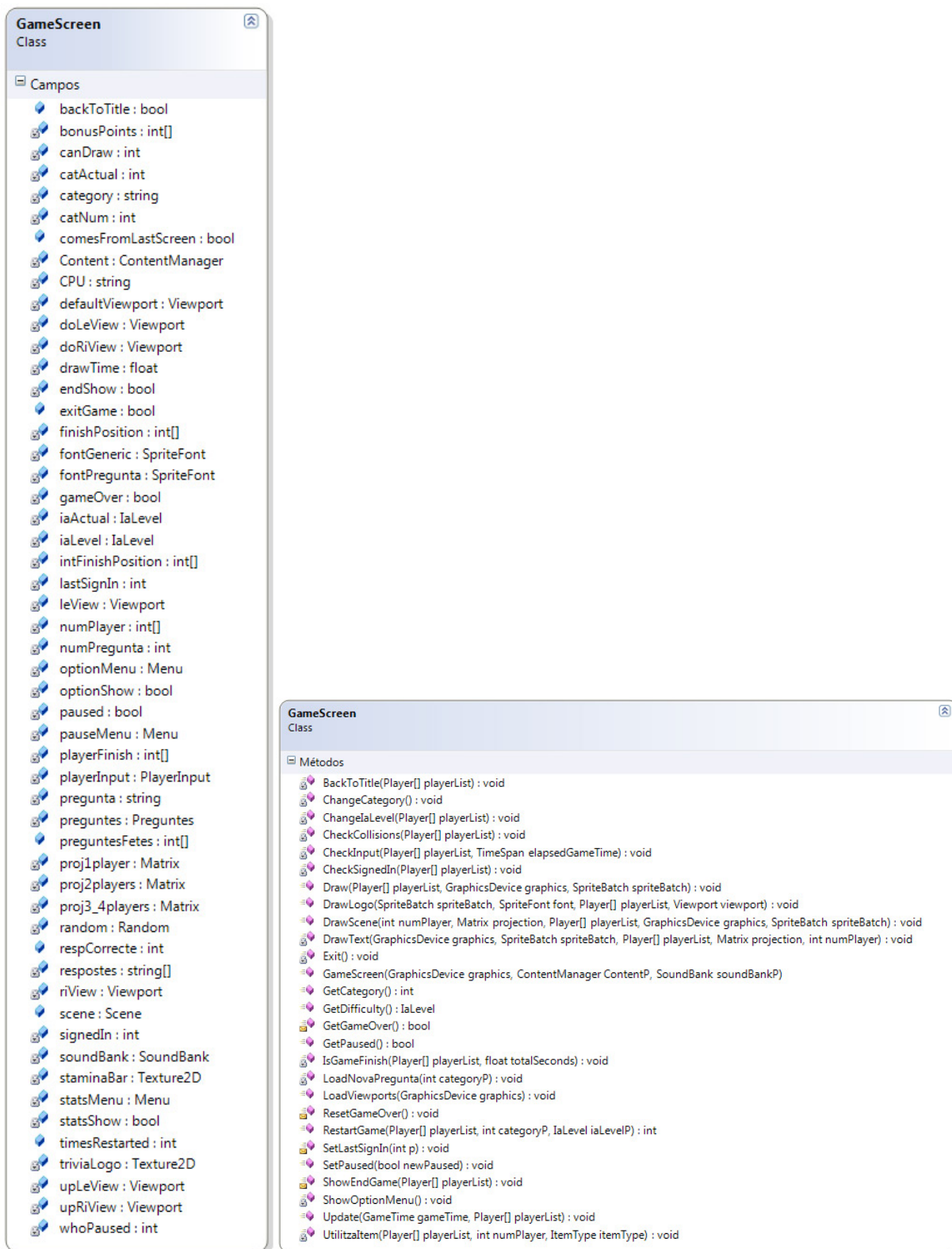


Figura 18: Classe GameScreen

Depenent dels jugadors que hagin iniciat la sessió, la pantalla estarà dividida en més o menys parts. Cada partició de la pantalla serveix per a què cadascun dels jugadors pugui seguir al seu personatge durant la partida:

- Si només hi ha un jugador identificat, aquest podrà utilitzar tota la pantalla per competir contra 3 personatges controlats per la IA del joc.
- Si hi ha dos jugadors identificats, la pantalla es parteix verticalment en dues parts iguals. Es va decidir utilitzar una partició vertical en comptes d'una partició horitzontal degut a que l'acció del joc transcorre la majoria del temps amb el personatge intentant arribar a dalt de tot de l'escenari, fent que sigui més important veure què té el personatge davant i darrere.
- Si hi ha tres o quatre jugadors identificats, la pantalla es parteix en quatre parts iguals. Si només hi ha tres jugadors identificats, la part inferior/dreta de la pantalla (que seria per al quart jugador) mostra un fons negre amb el logotip del joc.

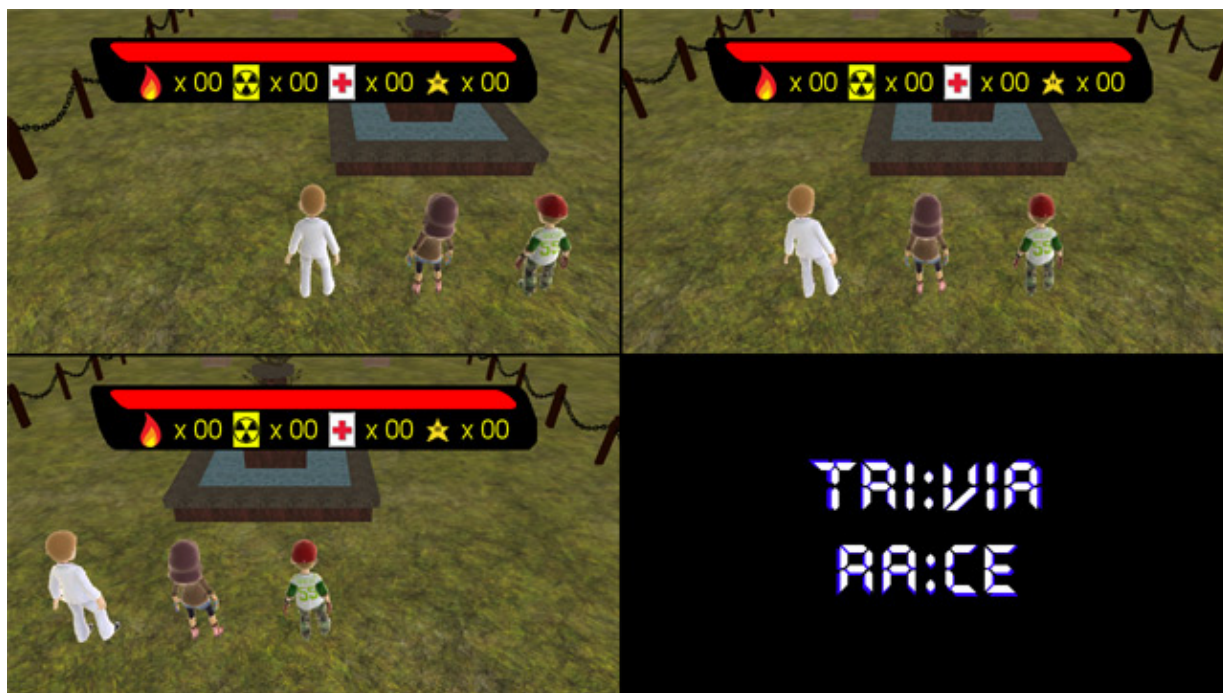


Figura 19: Pantalla GameScreen preparada per a tres jugadors

La pantalla de cada jugador mostra la interfície gràfica a la part superior on s'hi troben un seguit d'elements: L'element principal és una barra vermella horitzontal que indica gràficament l'energia que li queda al jugador. A mida que l'energia s'esgota (ja sigui perquè el jugador s'està movent per l'escenari o a causa d'un objecte utilitzat per algun dels altres personatges), la barra vermella es va reduint i es mostra una barra grisa. Si el personatge contesta una resposta, la barra canvia al color verd.

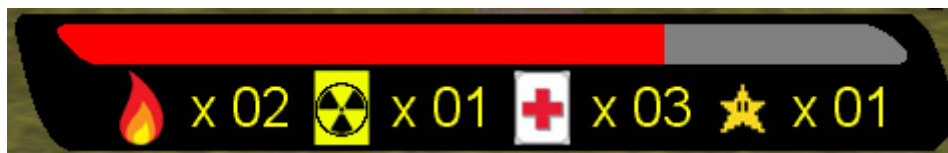


Figura 20: Barra de la interfície gràfica

Els altres elements que es mostren a la interfície són icones en miniatura dels objectes d'atac i defensa. A cada una d'elles les acompanya el nombre d'unitats que li queden al jugador. Aquest nombre s'actualitza en temps real a mida que utilitza cada un dels objectes, mostrant la informació de la instància de la classe Items de cada jugador. D'aquesta forma el jugador té una visió ràpida i clara de l'estat del seu inventari.

A més de la interfície gràfica, a la pantalla també es mostra l'escenari i l'avatar del jugador. Aquest avatar és el que el jugador ha dissenyat prèviament a l'hora de crear-se el compte guardat a la consola Xbox 360 i des del qual ha iniciat sessió.

Gràcies a que la plataforma per la que es desenvolupa és la Xbox 360 i aquesta permet utilitzar com a models de personatges els seus avatars, es va decidir fer-los servir com a personatges principals, agilitzant així la part de disseny de personatges. Com que aquests ja estan modelats en 3D i tenen moltes animacions predefinides, no s'hauria d'invertir tant de temps en el seu disseny.

Quan el jugador inicia la sessió dins del joc, aquest llegeix la informació referent de l'avatar i la representa gràficament durant la partida. Mentre dura el joc, la càmera va seguint l'avatar, que està situat a la part central/inferior de la pantalla. En el cas que el personatge estigui controlat per la IA, al no utilitzar cap compte de la consola, el joc escull de forma aleatòria totes les característiques gràfiques de l'avatar: sexe, alçada, color de pell, configuració facial i roba.

Els jugadors poden controlar als seus personatges mitjançant el joystick del comandament Xbox. Quan un jugador inclina la palanca, el joc llegeix el grau d'inclinació i la direcció en la que s'ha inclinat. Aquesta informació es llegeix mitjançant la classe Player i PlayerInput, i es reflecteix en si el personatge ha de córrer, caminar o simplement estar quiet, i en quina direcció ho ha de fer. Per utilitzar algun dels objectes que hagi comprat prèviament, el jugador ha de tenir com a mínim una unitat de l'objecte i ha de prémer algun dels quatre botons principals del comandament Xbox:

- Si es prem el botó A, es realitza l'Atac 1.
- Si es prem el botó B, es realitza l'Atac 2.

- Si es prem el botó X, es realitza la Defensa 1.
- Si es prem el botó Y, es realitza la Defensa 2.

Si el jugador no disposa de cap unitat de l'objecte que intenta fer servir, el joc simplement ignora l'ordre d'utilitzar-lo. Si té com a mínim una unitat, pot gastar-lo.

Els personatges controlats per la IA també poden utilitzar els objectes que han comprat prèviament a la tenda. Els criteris que segueix la IA per utilitzar els objectes depèn del nivell amb la que s'ha configurat:

- Si està configurada amb nivell fàcil, cada cop que es cridi el mètode `laWhatAction()` de la classe `Player`, el personatge té un 5% de possibilitats d'utilitzar un Atac 1, sempre que hi hagi una màxima distància entre ell i un altre personatge.
- Si està configurada amb nivell normal, el que primer comprova la IA és l'energia que li queda al personatge. Si té més de 15 punts d'energia, el personatge té un 5% de possibilitats d'utilitzar un Atac 1 (sempre que hi hagi una distància màxima entre ell i un altre jugador). Si no el fa servir, llavors té un 3% de possibilitats d'utilitzar un Atac 2. Si al personatge li queden menys de 15 dels 100 punts d'energia, té un 3% de possibilitats d'utilitzar una Defensa 1 per curar-se.
- Si està configurada amb nivell difícil, primerament també comprova l'energia restant. Si té més de 20 punts d'energia, la possibilitat que utilitzi un Atac 1 és del 5%, sempre que hi hagi una distància màxima. Si no el fa servir, llavors té un 5% de possibilitats d'utilitzar un Atac 2. En el cas que li quedin menys de 20 punts d'energia, primer té un 5% de possibilitats d'utilitzar una Defensa 1 i si no després un 5% de possibilitats d'utilitzar una Defensa 2.

Quan un jugador utilitza un objecte i ataca a algú, el jugador atacat rep punts de defensa, amb els que podrà comprar objectes de defensa quan tots els jugadors tornin a passar per la pantalla de la tenda. Pot ocórrer que el jugador gastí un Atac 1, que només ataca a les persones que estiguin dins d'un radi d'acció, i no hi hagi ningú dins d'aquesta àrea. En aquest cas, el jugador perd una unitat de l'objecte sense haver atacat a ningú. Com que l'Atac 2 afecta a tots els demés contrincants sense importar la distància a la que es trobin, aquest problema no es pot donar.

Un cop comença la partida, es mostra un gran cartell a la part central de la pantalla amb la pregunta de la ronda, i després sobre cada avatar apareixen el

nom del jugador que el controla. Si el personatge el controla la IA, el nom que apareix sobre l'avatar és "RandomAvatar".



Figura 21: Pantalla GameScreen amb els noms de cada jugador

Les preguntes que es mostren estan organitzades mitjançant la classe Preguntes.

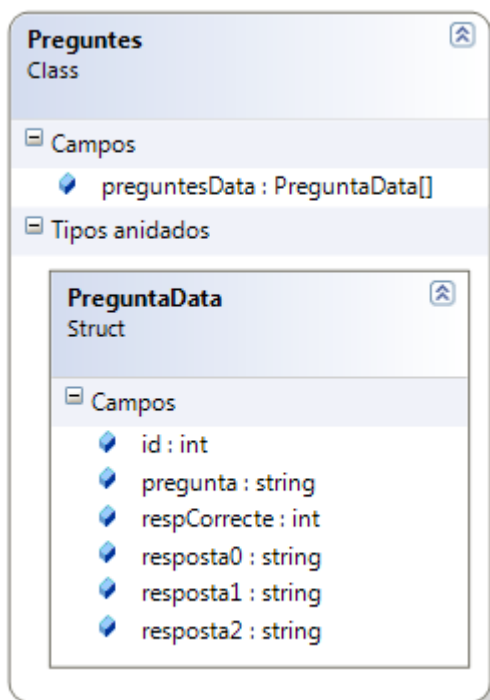


Figura 22: Classe Preguntes

Aquesta classe només conté l'estructura `PreguntaData`, que prepara els camps d'informació necessaris per llegir els arxius XML que contenen les preguntes. Llavors, la classe `GameScreen` s'encarrega de llegir l'arxiu XML de la categoria seleccionada a `MenuScreen`. A mida que es va llegint l'arxiu seleccionat, `GameScreen` omple les variables mitjançant l'estructura de `PreguntaData`. Aquesta organització de les preguntes en arxius XML facilita que es pugui ampliar el nombre de preguntes sense haver de tocar el codi del joc, ja que qualsevol persona amb accés al codi podria modificar l'arxiu XML i llavors només s'hauria de tornar a compilar i pujar-lo al Bazar de Microsoft com a actualització del joc.

A partir del moment en que surt la pregunta, tots els personatges poden començar a moure's per arribar abans que cap altre a la meta. Els jugadors han de controlar diverses coses alhora: el que estan fent els altres jugadors, els obstacles repartits per tot l'escenari i l'energia del personatge.

Aquesta energia es modifica segons l'acció que estigui fent el personatge. Si està corrent l'energia disminueix ràpidament, i si es mou caminant l'energia també disminueix, però més lentament. L'única forma per recuperar aquesta energia és estar parat o utilitzar objectes de defensa. Si l'energia del personatge arriba a zero, aquest es desmaia i es queda immobilitzat durant uns segons sense poder moure's ni utilitzar objectes d'atac o de defensa. Passat aquests segons, el personatge recupera uns punts d'energia i pot continuar endavant.

Els escenaris pels que es mouen els personatges estan gestionats a través de la classe `Scene`.

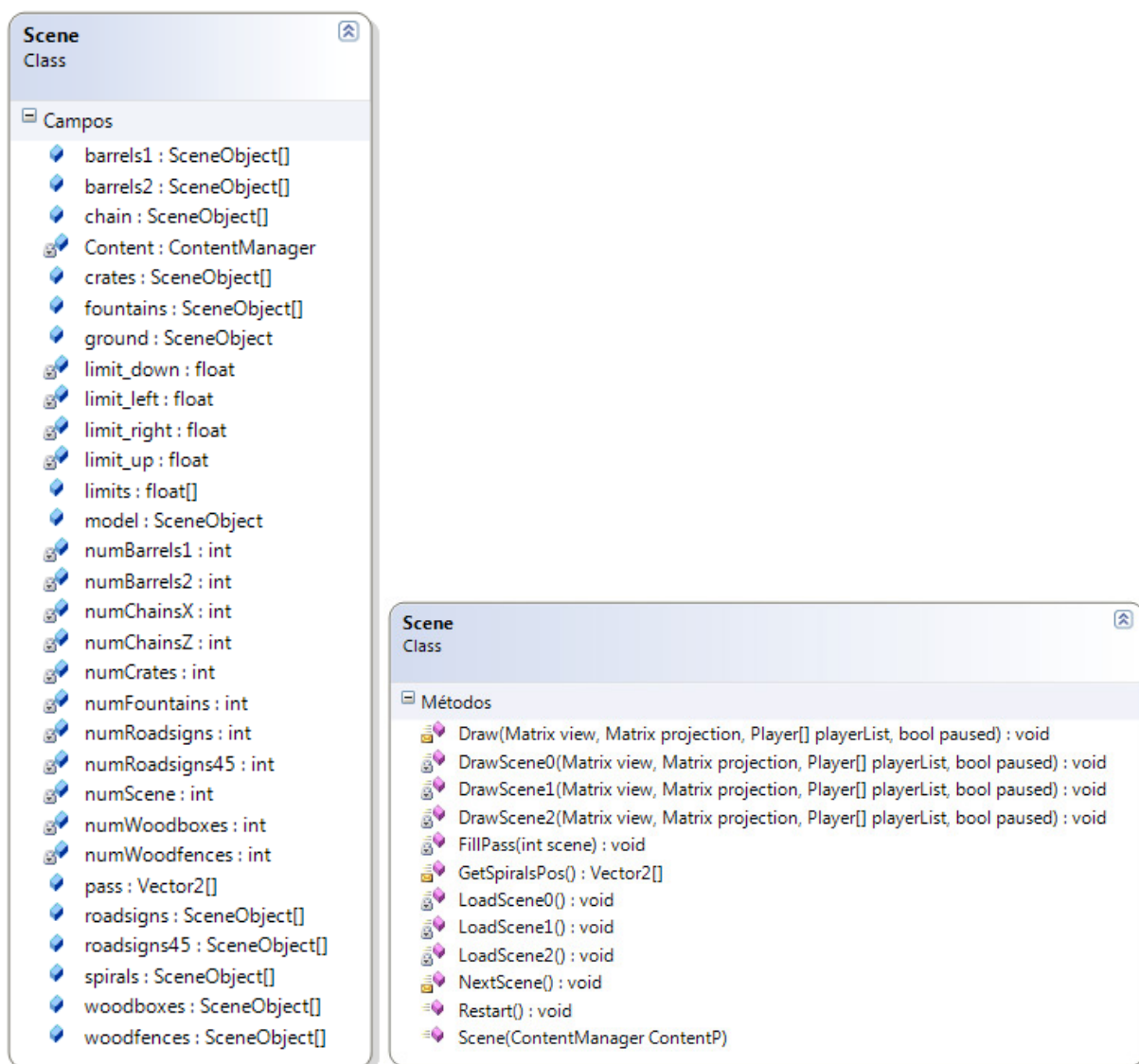


Figura 23: Classe Scene

Aquesta classe gestiona on han d'anar els obstacles en cada un dels tres escenaris que hi ha creats, així com els forats que hi ha entre els obstacles per on han de passar els personatges controlats per la IA. La classe també controla quin és l'escenari que s'ha de mostrar en cada moment i el pas d'un escenari al següent quan s'acaba la ronda.

L'escenari pel que es mouen els personatges està limitat per quatre tanques que formen un gran rectangle vertical, amb varis objectes pel mig. Aquests objectes serveixen d'obstacles als personatges perquè no puguin arribar al final de l'escenari simplement corrents en línia recta. Per a què els objectes es mostrin amb unes proporcions correctes a la pantalla s'utilitza la classe SceneObject.

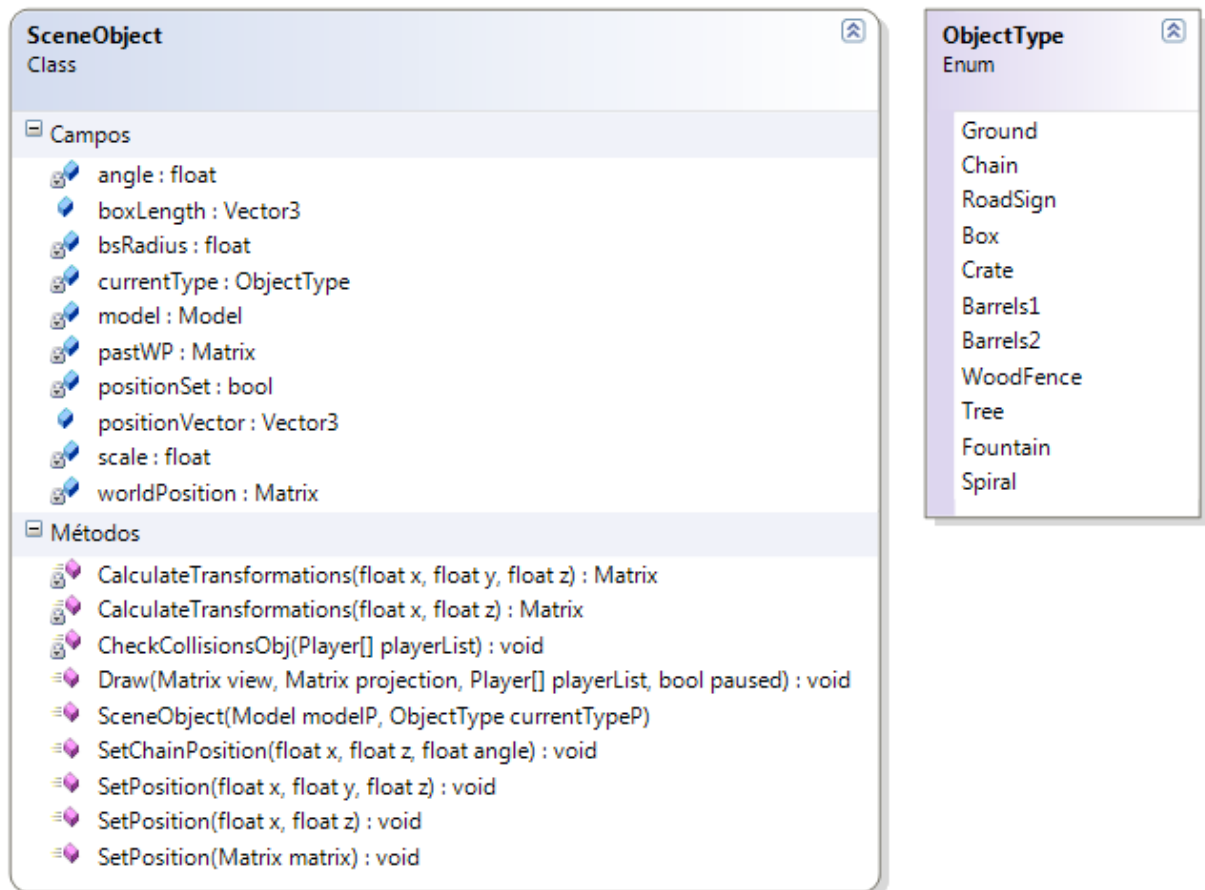


Figura 24: Classe SceneObject

Aquesta classe també conté l'estructura ObjectType, que ajuda a enumerar els tipus d'objectes que hi ha preparats per al joc. Amb SceneObject el joc pot accedir a la biblioteca d'obstacles que es poden posar en un escenari i saber exactament quines transformacions s'han d'aplicar al model en 3D perquè es mostri correctament proporcionat. Per no haver d'invertir molt de temps modelant els objectes en 3D, s'han utilitzat objectes ja modelats de les pàgines archibase.net i turbosquid.com, lliures de copyright, i únicament s'han hagut d'aplicar textures als models descarregats de les mateixes pàgines web.

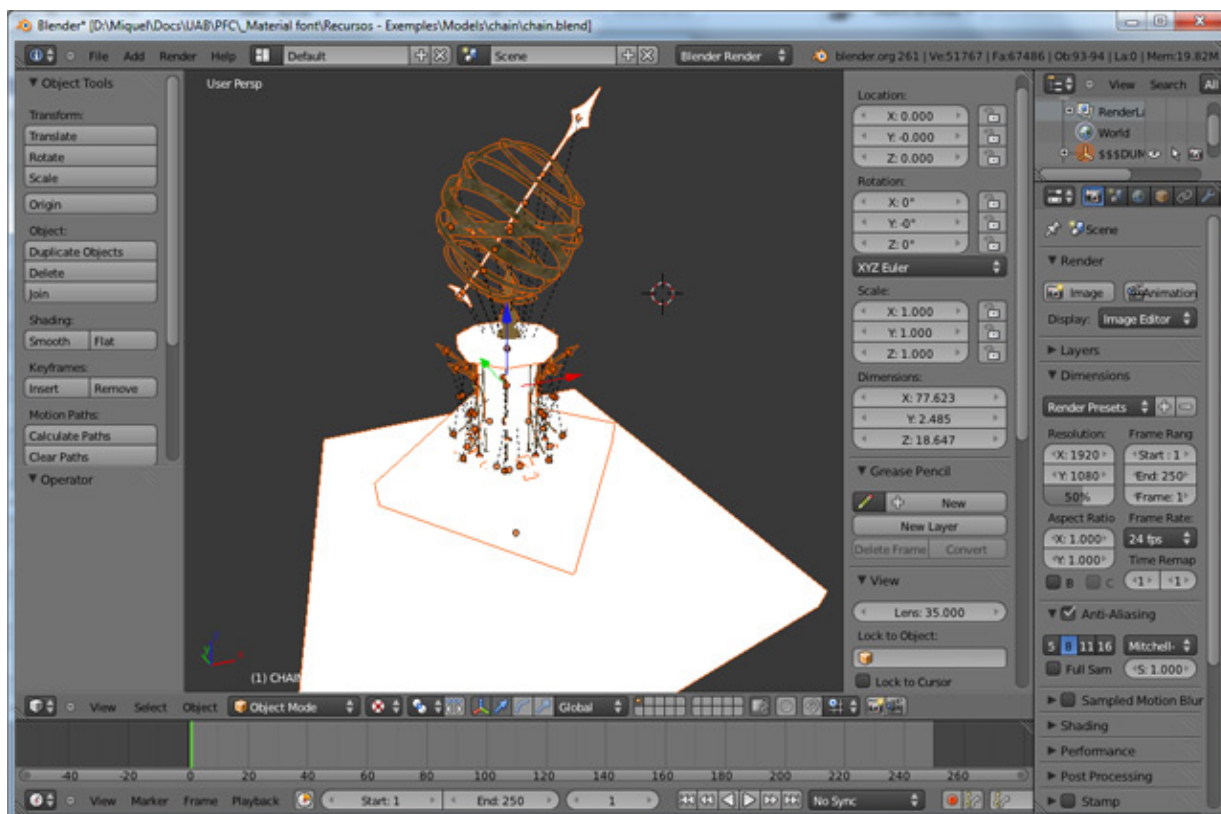


Figura 25: Model 3D de font en el programa Blender

Durant la carrera cap a la meta, els jugadors han d'intentar evitar els atacs dels demés mentre ells intenten que els altres s'endarrereixin. La IA, per la seva banda, intenta fer el mateix. Segons el nivell de la IA amb el que s'estigui jugant, els personatges controlats per aquesta aniran per un camí més o menys curt fins a la meta, administraran millor o pitjor els objectes que hagin comprat a la tenda, i un cop arribin a la zona de respostes, respondran més o menys bé la pregunta.

Per aconseguir que la IA pugui escollir un camí, a l'escenari s'han afegit a més dels objectes, uns punts claus que defineixen els forats entre els objectes per on poden passar els personatges, creant un sistema de grafs. La IA llegeix les coordenades dels forats més propers que té per endavant, i calcula la distància entre aquests forats i el personatge. El millor camí que pot escollir la IA és sempre escollint els forats que tinguin la distància més curta respecte el seu personatge.

Segons el nivell que hi hagi configurat per a la IA, aquesta escull el forat més adequat seguint aquestes regles:

- Si la IA està configurada en nivell fàcil, el personatge escollirà el camí de forma aleatòria.
- Si la IA està configurada en nivell normal, el personatge té un 50% de possibilitats d'escollir el forat amb la distància més curta.

- Si la IA està configurada en nivell difícil, el personatge sempre escull el forat amb la distància més curta.

Mentre dura la partida, qualsevol dels jugadors pot prémer el botó Start del comandament Xbox i activar la pausa. Quan s'activa la pausa, el joc queda congelat temporalment i apareix un menú. Només el jugador que ha activat la pausa pot moure's pel menú. Des d'aquest menú es pot tornar a la partida a través de la primera opció o tornant a prémer el botó Start. La segona opció mostra un submenú on es poden modificar les opcions de la categoria de les preguntes i el nivell de la IA. Si es canvia alguna d'aquestes dues opcions, la ronda torna a començar. De les dues opcions que queden del menú de pausa principal, la primera serveix per tornar a la pantalla de títol, reiniciant la partida i els punts acumulats, i la segona serveix per sortir del joc.

Al final de l'escenari, s'hi troba la zona de respostes. Aquesta zona està composta per tres grans espirals que delimiten la zona on s'ha de contestar. Cada espiral porta un cartell amb una de les respostes a la pregunta que es formula. Per contestar, els personatges s'han de posar dins de l'àrea de l'espiral que considerin que conté la resposta correcta.



Figura 26: Zona de respostes de GameScreen

Un cop un personatge s'hagi posat dins d'una de les tres espirals i no surti d'allà en cinc segons, el joc entén que aquella és la resposta que el jugador considera

correcta. Llavors, el personatge queda immobilitzat i el jugador ha d'esperar que els demés contestin.

La IA dependrà del nivell amb la que hagi estat configurada per escollir millor o pitjor la resposta:

- En el nivell fàcil, la IA escollirà de forma aleatòria la resposta.
- En el nivell normal, la IA té un 50% de possibilitats d'escollir la resposta correcta i un 50% de possibilitats d'escollir la resposta de forma aleatòria.
- En el nivell difícil, la IA sempre escollirà la resposta correcta.

Quan tothom ha escollit alguna de les respostes, la ronda s'acaba i es mostren els resultats. En el cartell s'indica el nom dels quatre jugadors, si han contestat o no correctament, i en quina posició han quedat respecte als demés. Els jugadors que hagin contestat correctament reben punts d'atac. La quantitat de punts d'atac que guanya cada jugador depèn de la posició en que hagin contestat. Com més aviat hagi escollit un jugador la resposta correcta, més punts d'atac s'endurà.

Un cop s'acaba la repartició de punts, qualsevol dels jugadors pot prémer el botó A del comandament i donar pas a la següent ronda de la partida amb una nova pregunta des del principi de l'escenari. D'aquesta manera torna a començar la cursa.

Cada ronda té un escenari amb una col·locació diferent dels obstacles. Quan s'han acabat tres rondes, es considera que el joc s'ha acabat i es fa un recompte final. En aquest moment apareix un cartell que mostra el nom dels quatre jugadors, en quina posició han quedat a cada una de les rondes i quants punts d'atac han aconseguit en tot el joc. Llavors, si un jugador prem el botó A, la pantalla torna a mostrar la tenda d'objectes, i tothom pot aprofitar els punts d'atac i defensa que han guanyat durant la partida per comprar més objectes per a la propera. Els punts es mantenen d'una partida a la següent, així que es poden jugar tantes partides seguides com es vulgui.

5. Proves i errors

En aquest apartat es descriuen les proves que s'han dut a terme per assegurar un bon nivell de qualitat del videojoc i els errors més importants que han sorgit durant el seu desenvolupament.

5.1 Proves

Com que el desenvolupament del videojoc es va planificar sobre una sèrie de fites, les proves més importants sempre s'han realitzat un cop es tenien els objectius per a la següent fita aconseguits. Cada classe o tros de codi nou que s'afegien eren provats per comprovar que funcionaven correctament.

Al principi, les proves es realitzaven sobre els petits projectes de prova que més endavant s'unirien per formar les primeres versions alpha del projecte. A partir d'aquí, les proves eren més exhaustives ja que no només s'havia de provar que el codi nou funcionés correctament, si no que la integritat del projecte continuava sent correcta.

Un petit grup de betatesters ha ajudat durant tot el desenvolupament a fer crítiques constructives sobre els progressos que s'anaven realitzant, així com consells per a properes fites. Això ha estat molt útil a l'hora de tenir altres punts de vista del projecte que no s'havien tingut en consideració.

Malauradament, la naturalesa de XNA per a Xbox no permet la distribució del joc a altres consoles que no siguin membres del Creator's Club. És per això que aquesta fase s'ha dut principalment des de la consola en la que es portava el desenvolupament, però també es volia arribar al màxim nombre de persones possible. Per aquest motiu ja des del principi del projecte s'ha anat creant un diari online que recull els progressos que es duen a terme amb el projecte. Aquest blog compta amb vídeos i imatges que mostren el dia a dia del desenvolupament. Les entrades del diari es recullen a l'annex II.

El grup de betatesters durant la fase final del desenvolupament també ha provat directament el joc de la consola de desenvolupament i ha buscat errors que no s'haguessin tingut en compte durant la fase de programació.

5.2 Errors durant el desenvolupament

Com a qualsevol altre desenvolupament de software, a mesura que aquest s'ha anat realitzant, han anat sorgint molts errors que s'han anat corregint. La majoria d'aquests han estat fàcils de corregir, però hi ha hagut alguns que han portat més feina. En aquest apartat es tractaran aquests errors i s'explicarà com es va procedir per corregir-los.

El primer error important del projecte va sorgir durant la programació de les múltiples càmeres que el joc havia de controlar quan hi havia més d'un jugador amb la sessió iniciada.

Amb només un jugador la càmera seguia correctament al personatge que tocava. En canvi, quan hi havia més d'un jugador alhora, cada una de les particions de la pantalla que havia de seguir únicament a un dels jugadors prenia com a referència tots els jugadors que hi haguessin connectats, provocant una superposició dels avatars. Si es desactivava la càmera de seguiment i s'activava una genèrica, els avatars es movien correctament per l'escenari.

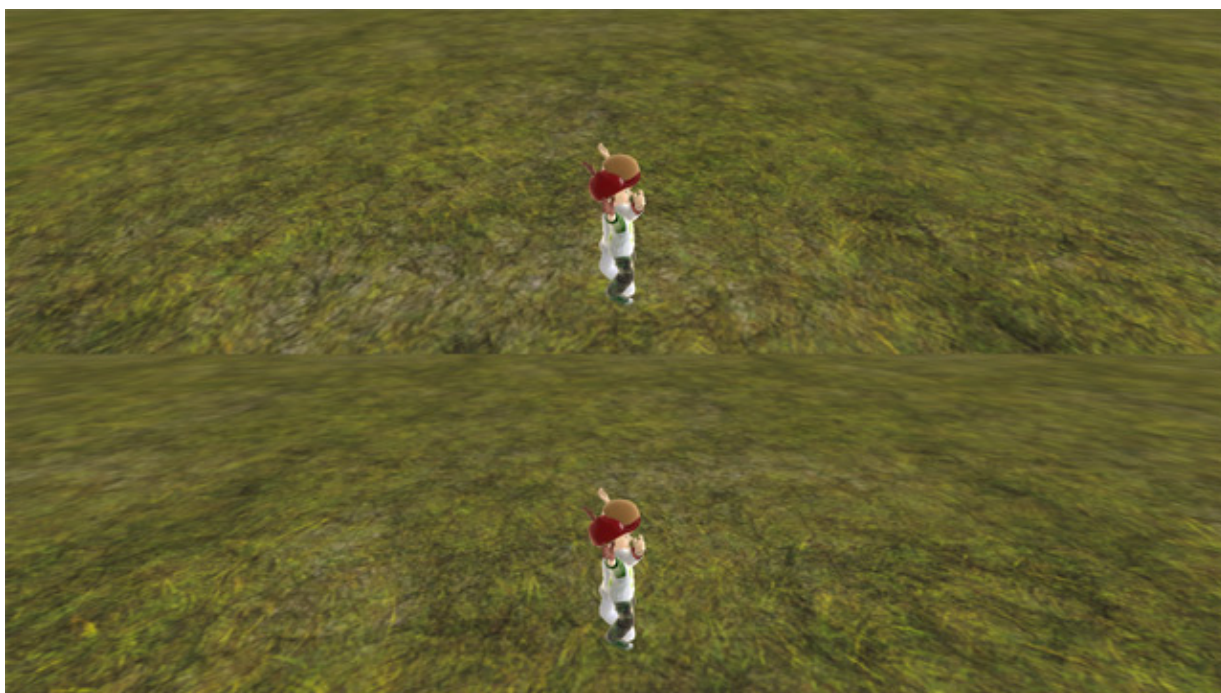


Figura 27: Pantalla d'error de gestió en les càmeres multijugador

L'error es trobava a l'hora de decidir quin personatge s'havia de mostrar en cada partició de la pantalla. Tot i que la partició 1 havia de mostrar al personatge 1 i la partició 2 al personatge 2, el codi feia que quan una partició preguntava quin personatge havia de mostrar, se li passaven totes les càmeres dels personatges

amb la sessió iniciada, en comptes de la del personatge que realment tocava, i per això cada partició mostrava tots els personatges com si estiguessin a la mateixa posició de l'escenari. Un cop corregida la càmera que es passava a cada partició de la pantalla per mostrar-la, no va donar més problemes.

El sistema de col·lisions entre personatges i objectes també va donar molts problemes durant el desenvolupament. Primerament es va intentar crear un sistema de *bounding boxes* i *bounding spheres* per a cada personatge i objecte de l'escenari, depenent de la seva forma. Tot i que al principi semblava que funcionava correctament, algunes vegades els personatges quedaven enganxats al col·lisionar entre ells, o el radi de col·lisió era massa gran o massa petit i les col·lisions es produïen massa d'hora o massa tard.

Això va portar al muntatge d'un sistema de col·lisions molt més complex, personalitzant cada un dels objectes amb el seu *bounding box*, però que no donava els resultats esperats i portava massa feina, amenaçant la planificació.

Per aquest motiu, es va decidir dur a terme un sistema de col·lisions més rudimentari, que tot i que no era tant fi com es desitjava, funcionava correctament. Aquest sistema consistia en marcar uns límits perifèrics a cada jugador i objecte. Quan el pròxim moviment que han de fer dels punts centrals d'aquests límits coincidien a l'espai, es cancel·lava l'última acció del personatge que havia intentat fer. D'aquesta manera, els personatges no arriben mai a sobreposar-se entre ells o contra altres objectes.

L'input dels comandament també va donar més d'un error. Quan s'estava provant el joc, es va veure que de totes les vegades que se li demanava al personatge que utilitzés algun objecte, algunes d'elles no es realitzaven. Aquestes accions no seguien un patró a l'hora de fallar, si no que fallaven aleatòriament. Això va portar a realitzar una cerca molt exhaustiva tant de la part d'utilització dels objectes, com de la lectura de l'input del comandament.

Al final es va veure que la classe `Player`, que en aquell moment era l'encarregada de llegir l'input dels comandaments, no podia realitzar de forma àgil totes les comprovacions que se li demanaven a cada instant degut a una saturació de lectures. Per això, es va decidir aïllar la part que llegia l'input dels comandaments en una nova classe, anomenada `PlayerInput`. A la que es va aïllar aquesta part de codi, l'input va deixar de fallar.

L'últim problema important del projecte va sorgir cap al final del desenvolupament, tot i que aquest cop es va donar per culpa de factors externs. Les llicències per a estudiant que permeten desenvolupar XNA per a Xbox 360 són anuals, i justament la llicència que s'utilitzava per desenvolupar el projecte va caducar a mitjans d'abril. Encara que aquestes llicències anuals valen 75€, Microsoft renova als estudiants les llicències de forma gratuïta, però s'han de seguir uns passos molt concrets.

Tot i que no hi hauria d'haver problema per a reactivar el compte, al finalitzar els passos aquest no es va activar com hauria. A l'escriure a Microsoft explicant el problema, en comptes de donar una solució, contestava dient que el departament al que havia escrit no se'n podia fer càrrec, i que provés a escriure a un altre. Microsoft es va anar passant la pilota entre varis dels seus departaments, com el departament d'App Hub per a XNA, el de Creator's Club per al desenvolupament per a Xbox 360 i el de DreamSpark per a les llicències d'estudiants. Tot i escriure a cada departament tal i com recomanaven ells, cap volia fer-se càrrec del problema ni sabia trobar una solució.

Al cap de 10 dies d'intercanvi de correus electrònics, i a punt de crear un altre compte pagant els 75€ per continuar el desenvolupament del projecte, la llicència d'estudiant es va reactivar, tot i que cap dels departaments va comunicar què és el que havia fallat.

6. Conclusions

6.1 Compliment d'objectius

Treballar mitjançant iteracions i fites ha permès una flexibilitat a l'hora de donar forma al projecte i marcar els objectius principals. D'aquesta forma s'han escollit els objectius més importants al principi del desenvolupament i s'han anat afegint alguns més a mesura que aquest avançava i quedaven clares les possibilitats que oferia XNA. Al marcar uns objectius realistes mitjançant les iteracions, es pot dir que el projecte ha estat un èxit per poder accomplir la gran majoria.

La majoria d'objectius que no s'han pogut dur a terme s'havien marcat com a objectius al principi del desenvolupament degut a la falta de coneixements a l'hora de saber les possibilitats que ofereix XNA, però s'han acabat descartant després d'investigar com si es podien dur a terme. Per exemple, durant la fase d'investigació, una de les propostes inicials per al projecte era crear un portal web en el que es poguessin introduir noves preguntes i respostes. Aquest portal, mitjançant una base de dades, permetria actualitzar els arxius XML que contenen els qüestionaris del projecte molt fàcilment, ajudant a ampliar l'actual qüestionari.

Aquesta idea però, es va haver de descartar degut a les pròpies limitacions tècniques de XNA, que només permeten una connexió a través d'Internet amb altres consoles Xbox 360, i per tant la connexió amb el servidor del portal web quedava descartada. Per poder dur a terme aquesta idea, l'única possibilitat era enviar les noves preguntes a algú amb accés al codi, i que aquesta persona tornés a compilar el joc i el pugés al Bazar Xbox Live com una actualització. Tot i que la idea es podia dur a terme es perdia la facilitat d'actualització, que era el seu objectiu principal.

6.2 Desviacions

El projecte no ha sofert grans desviacions, ni temporals ni d'objectius. Marcar fites amb objectius per a cada reunió va ajudar a fer una planificació temporal més realista, aconseguint complir la pràctica totalitat d'elles.

Tot i això, hi ha hagut algunes idees que no s'han dut a terme degut a la falta de temps, com la de recrear escenaris més complexos. Aquestes idees es van descartar a l'hora de prioritzar altres objectius més necessaris per donar per finalitzat el projecte, i van quedar com a possibles línies d'ampliació.

6.3 Línies d'ampliació

Un projecte d'aquest tipus sempre es pot millorar d'alguna forma o altra, ja sigui afegint-hi més característiques o millorant les ja existents. Malauradament, la limitació temporal del projecte fa que moltes d'aquestes millores no es puguin dur a terme i s'hagin de prioritzar uns objectius sobre uns altres. Si el projecte pogués disposar de més temps, unes ampliacions molt interessants serien:

- Joc online: L'apartat multijugador del projecte només permet jugar des de la mateixa consola. Amb el joc online, cada jugador podria jugar des de casa seva amb els seus amics o amb gent de tot el món.
- Més categories i preguntes: El joc presentat no deixa de ser un projecte, i com a tal conté les preguntes i categories suficients per demostrar la idea bàsica. Si es volgués comercialitzar el joc, s'hauria d'ampliar molt el catàleg de preguntes i categories, ja que és la base del joc.
- Personalitzar les categories: Una ampliació interessant podria ser una llista de categories de preguntes personalitzable. Una idea així es podria dur a terme per exemple si els jugadors poguessin barrejar preguntes de diverses categories per crear-ne una de nova, puntuar les preguntes per saber quines agraden més, o personalitzar la freqüència amb la que es vol que torni a aparèixer una pregunta (més sovint, igual, menys sovint, mai més).
- Més objectes a la tenda: Una major quantitat d'objectes permetria crear més estratègia a l'hora de competir. Per exemple, es podrien afegir objectes que modifiquessin la velocitat a la que es mouen els personatges, fent córrer més al propi personatge o alentint la marxa dels contrincants.
- Escenaris més vius: Els escenaris del projecte contenen obstacles que només serveixen per dificultar el pas als personatges, però són obstacles estàtics. Seria interessant poder afegir objectes més dinàmics que fessin més interessant la cursa, com ara trampes amagades que aturin als personatges, forats al terra per on poden caure els jugadors, transportadors que et fan avançar o retrocedir grans distàncies o objectes que impedeixen el pas de forma intermitent.
- Una IA més avançada: La IA d'aquest projecte segueix unes regles molt bàsiques, i per tant seria molt interessant afegir-ne més, de forma que el seu comportament s'assemblés més a la d'un jugador real. També seria interessant poder ajustar un nivell de IA individualitzat per a cada personatge que controlés.

6.4 Conclusions finals

Arribats a aquest punt del projecte, l'últim pas que li quedaria al joc seria comercialitzar-lo. Per fer-ho, primer hauria de passar una revisió per part de Microsoft i després una segona revisió per part de la comunitat d'usuaris d'App Hub. Si el joc passés les dues revisions, ja es podria posar a la venda al Bazar de Xbox Live. Tot i la feina que ha portat desenvolupar aquest videojoc, no deixa de ser un projecte de final de carrera al que li faltaria ampliar el seu contingut per arribar als límits de qualitat estàndards del mercat actual, i per tant aquest punt queda pendent per un futur.

Tanmateix, he quedat molt satisfet i molt content de la feina feta, i crec que he complert amb èxit els objectius que vaig plantejar-me a l'iniciar aquest projecte, tant de feina com els personals. Els videojocs sempre han sigut una de les meves aficions, i endinsar-me una mica en aquest món m'ha servit per veure la gran quantitat de feina que poden portar fins i tot els detalls més petits. Ha estat aquest procés de descobriment d'una nova faceta dels videojocs el que m'ha mantingut motivat durant tot el projecte i m'ha animat a continuar millorant dia a dia el joc, buscant noves característiques i polint les ja existents.

Gràcies a aquest projecte he pogut dur a la pràctica molts dels coneixements que he après durant aquests anys de carrera, com la necessitat de tenir les idees ben plantejades abans de desenvolupar res; a més d'ajudar-me molt la base de coneixements rebuda a l'hora d'aprendre nous llenguatges de programació, com en aquest cas ha estat XNA i C#.

6.5 Agraïments

Vull donar les gràcies a aquelles persones que m'han guiat durant el projecte i m'han donat noves idees i bons consells. Per començar, he d'agrair al meu tutor Xavier Verge per guiar-me en el desenvolupament del projecte i ajudar-me a portar una planificació més realista i assequible. També volia donar les gràcies al Dr. Enric Martí pels bons consells que em va donar per millorar el joc i per remarcar els punts que eren més importants per la memòria. Agrair també a totes les persones que m'han ajudat a provar el joc, han fet crítiques constructives i han aportat noves idees. Per últim als meus pares, al meu germà i a l'Eva, pel suport moral i els ànims tan en els moments bons com en els que no sortien els resultats esperats.

7. Bibliografia

- **JAMES, SEAN** (2010). 3D Graphics with XNA Game Studio 4.0. Packt Publishing.
- **MILLER, TOM** i **JOHNSON, DEAN** (2011). *XNA Game Studio 4.0 Programming*. Addison-Wesley.
- **MILLINGTON, IAN** i **FUNGE, JOHN** (2009). *Artificial Intelligence for Games -Second Edition-*. Morgan Kaufmann Publishers.

Fonts electròniques:

- **GROOTJANS, RIEMER**. *Riemers XNA Tutorial*. Direcció electrònica: www.riemers.net
- **MICROSOFT**. *App Hub – windows phone and xbox live indie games development*. Direcció electrònica: <http://create.msdn.com>
- **MICROSOFT**. *Microsoft DreamSpark*. Direcció electrònica: www.dreamspark.com
- **MICROSOFT**. *MSDN – Explore Windows, Web, Cloud, and Windows Phone Software Development*. Direcció electrònica: <http://msdn.microsoft.com>
- **PLACIDA, DANIEL**. *Architectural Home Design, 3D models, Quality Textures. Online Interior Design*. Direcció electrònica: www.archibase.net
- **TURBOSQUID**. *3D Models, 3D Modeling Textures and Plugins at TurboSquid*. Direcció electrònica: www.turbosquid.com
- **XNA WIKI**. *XNAWiki*. Direcció electrònica: www.xnawiki.com

8. Annexos

8.1 Annex I: Manual d'usuari

Per jugar al videojoc TriviaRace es necessita com a mínim un comandament de Xbox 360. Durant el joc, es fan servir dos tipus de configuracions fàcils de recordar:

- Controls durant les pantalles de títol, opcions i tenda:



- Controls durant la carrera:



8.2 Annex II: Blog del projecte

Paral·lelament al projecte i quasi bé des dels seus inicis, es va crear un blog que permetés dur un seguiment a nivell personal del projecte i veure la seva evolució en el temps. Aquest seguiment havia d'ajudar a l'hora d'escriure la memòria del projecte, ja que es disposaria d'un registre escrit i gràfic dels progressos més importants. Cada cop que s'escrivís una entrada nova, havia d'estar acompanyada d'una imatge o vídeo que mostrés les novetats del desenvolupament.

Aprofitant que es portaria aquest registre del projecte, per poder arribar a tanta gent com fos possible es va decidir escriure'l en anglès i etiquetar cada entrada amb títols com "XNA" o "avatars", de forma que gent de tot el món interessada en projectes escrits en XNA pogués trobar i visitar el blog i donar la seva opinió. D'aquesta forma, la idea de blog personal va anar evolucionant més cap a blog de desenvolupament, on la gent participés i digués la seva.

En aquest sentit el blog va ser tot un èxit, ja que sense publicitar-lo de cap manera, a dia d'avui recull entre totes les entrades un total de 36 persones que segueixen les publicacions del desenvolupament del joc. A més, diverses persones han escrit directament al compte de correu del blog per deixar comentaris, opinar sobre el videojoc i fer la seva crítica, rebent per tant un feedback molt positiu per al projecte.

La web del blog és la següent:

<http://bakaosaka.tumblr.com>

A continuació es poden llegir un extracte dels textos del blog i veure alguna de les imatges del desenvolupament. A la pàgina web s'hi troben els vídeos que mostren l'evolució del projecte.

11/10/11



19/11/11

This one [test] merges two default animations into one, waving with the right arm and celebrating with the rest of the body.



20/11/11

The avatars can now load custom animations (like running) and move through the world.



23/11/11

First of all today, a simple XML reader I wrote some days ago. This isn't as funny as playing with the avatars, but it's still needed for the project!

```
ID: 0
Pregunta: What is a man?
Resposta: A miserable little pile of secrets
ID: 1
Pregunta: What do you here?
Resposta: ....
```

Another test, now starting to implement the cameras for a future local multiplayer. Based on my first load_avatars test. I'm starting from near zero with this things, so it might take some time!



08/01/12

There have been some new updates, and the basics for my game are now established. Let's see the news!

First of all, a simple and nice test for the left thumbstick of the Xbox pad. This shows in a scale from 0 to 1 the percentage of the inclination of the thumbstick, and the supposed action the avatar has to make.

Using the leftThumb_test as the starting point, in this test I mixed it with a default animation (the standing pose) and some custom animations (the walking and running ones). Now I can make the avatar do whatever I want him to do!

Now I want to show you one of my biggest blocks since I started this project: The control of the cameras in the 3d world when there is more than one active player. When the second player signed in, the cameras showed the avatars in the same position, and moving one player seemed to move all the active players at once.

Well, that was because every avatar has his own camera located in his back, and every split screen was drawing all the active avatars from their own camera. That means that all the avatars were drawn in the same location. The solution for the problem was to draw all the players from only the viewport's player camera, so the avatars were drawn in the correct relative distances.

In this project I also implemented a dynamic partition of the screen based on the number of active players at the moment. It is seen in the video that at the start only one player is signed in, then comes a second and a third player, and the screen splits in two and then in four parts. Finally, the second player signs out, and so the screen is only splitted in two, showing the first and third player on the screen. The second player's avatar also disappears and in his place the game generates a random avatar.



At this point I started to put the tests together to make the basis of the project but then the main code became too large and complicated to manipulate, so I restructured it. In order to do this, first of all I made a simple menu test with three menus (titleScreen, menuScreen and gameScreen) being a project class and controlled from Game1.cs, the base of the project. I also splitted the properties of the player from the main code in a separated class.

Starting from titleScreen, when the Start button is pressed, the project stops drawing and updating titleScreen and starts doing it for menuScreen. Now the player can press A and load gameScreen or press B and go back to titleScreen. The same occurs with gameScreen.

These new classes let me divide the code in various classes, and make the main class (Game1.cs) a little lighter.



In this update, I modified the menuScreen a little. Now the non-signed in players can sign in pressing only the Start button. The menu screen also shows the gamertags for the signed in players. In this version of the code the pause in the gameScreen has also been implemented, but it's not seen in the video (my fault!). The avatars still don't collide with each other :(

At last, my final update (for now).

Now the project has a shopScreen that allows the players buy attack or defense items before the game is started. This screen is shown every 3 games. As it can be seen in the video, now the avatars collide with the other players (wohoo!).

The input code for the players has also been restructured so when a button is pressed and the game jumps into another screen, that screen doesn't function like the button is pressed. That gave me some headaches, because when you went from the menuScreen to the shopScreen (pressing the A button) and the shopScreen appeared, the input for Player 1 was still that the A button was pressed, and so the shop understood that the player wanted to buy an item. This problem appeared also when the player was in the shop screen and pressed the Start button to start the game, and the input was interpreted like the player wanted to pause the game.

12/01/12

New update! These days I've been working on the avatars' animations. Now the code allows including new animations without having to rewrite much of it.

The standing animations are now random and every time the game is reloaded or the standing animation finishes its movements it plays a new animation. The same happens when the new "throwing" animation that I've added finishes.

The last update I did to the code today is to merge the throwing animation with the walking and running ones, so the avatar can attack while running or walking. I used the past "multiple_animations" example and adapted it to the new animations to achieve this. The animations are also synchronized, so when the avatar is running he can launch an attack, and if he starts walking instead of running before the attack animation finishes, the attacking animation doesn't stop.



29/01/12

Big update! The field is not alone anymore! I've added some nice copyright-free 3d models to populate the game. The collision system has been implemented too. The code for the collision between players is recycled and it works fairly good for the objects. If I have some free time I will improve it (those camera shakes... :|).

The game has 3 rounds for every game. When the third round ends, the game goes back to the shop and the players can buy everything their points let them. In every round, the objects' placement changes to one of the three pre-designed scenarios.

The xmlread test has been also implemented. Now the game can read the questions from a XML (along with the true and false answers). In every game, one question is randomly selected, and the players can answer in the answer zone, at

the end of the field. The questions selected in every round will not appear again till the game ends, when the used questions array is reseted.

At the end of the field is the answer zone, where the players can answer to the questions. Every question has three possible answers, and only one of them is correct. In order to answer a question the player must enter into one of the three spirals, depending what answer wants (A, B or C) (the in-game text is not implemented yet!). When the player stays for more than 5 seconds in one of the three spirals, he/she cannot move anymore, and the game accepts his/her answer. The first to enter the correct answer wins more attack points! :D



13/02/12

New post, new update! This time the game shows the in-game test question, the players' gamertags and the three possible answers (in the answer zone, one answer inside each spiral).

I also created a menu class, which I used to make the pause menu. The class highlights the current position in the menu, and allows one or more players to navigate through it. The MenuScreen is also updated with this class and now the players can choose a question category for the test and the CPU level (not yet implemented!).

The shop is updated too. Now it loops for each signed in player and stores all the objects that the player buys with his/her points. These objects are now counted,

and once the game has started, the player can "fire" an attack for each attack object he/she has.

The last change made is that a player can join the game only before it starts. Once the game has started, if a player joins or leaves before the game ends, a message appears and the game goes back to the title screen.

27/02/12

The AI implementation has finally started!

Depending on the CPU level, the AI chooses one of the three answers and the path to go till the end of the scene. If the CPU level is set on Hard, the avatar will run through the shortest path, and choose the correct answer. On the other hand, if the CPU level is set on Easy, the avatar runs through a random path and chooses a random answer.

When the CPU arrives at the end of the stage and chooses an answer, the avatar animation changes from running to standing.

05/03/12

The stamina system is on! Now the player uses the stamina when the avatar walks and runs. If the stamina level runs out, the avatar faints and has to rest some seconds before it can run again. The only way to restore stamina is wait without moving for a seconds.

The same goes for the AI!



11/04/12

This time there are a lot of changes. To begin with, the bonus system is implemented for the players and the AI. When you choose the right answer, you win some attack points you can spend in the shop, and buy attack items. When any other player attacks you, you win some defense points. There are four different items:

- Attack 1: All the players in a short range from you lose some stamina power.
- Attack 2: All the players lose less stamina power than in the attack 1.
- Defense 1: You can recover some stamina power.
- Defense 2: You are invincible for some seconds.

At the end of the round, a screen shows the results (winners and positions).

The graphic changes are for sure the more visible. The menus have a new style and there are some new images for the items. The stamina bar has changed too and now shows how many items the player has left.

And of course, now we have music and sound effects too!



03/05/12

The game is almost complete! This time the only visible changes are the instructions screen in the first menu, the winner screen when the three game rounds are completed and the introduction of the real questions for the test. The stamina bar changes its color from red to green when the player has accepted an answer.

However, this time there are a lot of bug fixes with the collision and the animation systems, among other minor bugs.

Miquel Sitges Blasco
Sabadell, juny de 2012