



Sendas Pasadas

Josep Martí Orive Jimenez
Grau Multimèdia
Videojocs

Helio Tejedor Navarro i Jordi Duch Gavalrà
Joan Arnedo Moreno

18/01/2018



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Creació d'un videojoc: Sendas pasadas</i>
Nom de l'autor:	<i>Josep Martí Orive Jimenes</i>
Nom del consultor/a:	<i>Helio Tejedor Navarro i Jordi Duch Gavaldà</i>
Nom del PRA:	<i>Joan Arnedo Moreno</i>
Data de lliurament (mm/aaaa):	<i>01/01/18</i>
Titulació o programa:	<i>Grau Multimèdia</i>
Àrea del Treball Final:	<i>TFG-Videojocs</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Combat, espases, medieval</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball.</i>	
<p>Sendas Pasadas és un videojoc en 2D amb gràfics vectorials ambientat en l'època medieval. La intencionalitat principal és fer un videojoc realista i immersiu per el jugador, amb el qual hagi de fer servir la seva empatia per tal de millorar la seva relació amb el joc i la seva habilitat amb el mateix.</p> <p>Per fer això, fem servir el mínim de GUI possible i fem uns moviments i gràfics realistes en relació al món real, per tal de que l'usuari pugui sentir-se immers i no estigui pensant tota la estona que està jugant a un videojoc.</p> <p>Està programat amb C# en unity, fent servir totes les seves possibilitats en quant a animacions, programació i configuració.</p> <p>El resultat ha estat bastant positiu, en relació al que es volia fer, tot i que el procés ha estat molt costós i, per poder realitzar bé el gameplay, s'han hagut de deixar d'implementar altres funcionalitats afegides.</p>	

Abstract (in English, 250 words or less):

Sendas Pasadas is a videogame based on the medieval time, where the player will do with different functionalities, different empathic exercises.

That videogame is based on the empathy of the player to know what to do and how to do it better, feeling in the skin of our character and trying to think like they would in that situation.

To make that empathy, we try to avoid the GUI elements and try to make the player to play as he can with the elements inside the graphics.

The graphics try to be as realistic as possible and the possibilities and animations are the same as a normal person would have.

It is programmed with C# and its configured with Unity's interface, trying to exploit to the limits their possibilities to the 2D games.

The final result is satisfiable to the objectives we had and the gameplay is funny and realistic as can be a 2D vectorial game.



Índex

Objectius: pàgina 2

Conceptualització: pàgina 3

Disseny: pàgina 7

Implementació: pàgina 21

Conclusions: pàgina 28

Vídeo de presentació:

https://www.youtube.com/watch?v=uJb0ZLgAN6M&feature=youtu.be&ab_channel=Animimar

Objectius

Sendas pasadas és el resultat de l'esforç, la determinació i, sobretot, del meu amor pels videojocs desde que tinc memòria. Durant la meva vida he jugat molts videojocs, alguns millors que d'altres, però alguns pocs d'ells m'han arribat a endinsar en el seu món i tenir-me meravellat amb la història que explicaven, com feien servir els recursos d'interacció dels que només poden fer servir els videojocs i, sobretot, passant-ho bé, encara que estigués passant per un mal moment a la meva vida. Per això, el meu objectiu aquest treball de fi de grau era fer un videojoc amb un impacte similar, una obra que es pogués considerar "art" per mi, amb tot l'esforç que fes falta. A arrel d'aquest objectiu principal van sorgir la resta d'objectius que volia accomplir amb aquest treball:

- Aprendre a fer servir Unity.
- Aprendre a programar en `c#` i amb un codi orientat als videojocs.
- Millorar els meus dibuixos i animacions amb tauleta gràfica i ratolí.
- Aprendre com fer una bona narrativa cohesionada i amb un bon missatge.
- Aprendre tot el procés d'un videojoc, desde la seva concepció fins la seva publicació.
- Fer un videojoc del que sentir-me orgullós més enllà del treball de fi de grau i la seva nota.
- Crear un sistema de joc innovador i original.
- Orientar el videojoc cap a l'empatia i com el jugador es relaciona amb ell amb els mínims objectes per pantalla.
- Passar-ho bé fent el que m'agrada.

Sobre aquests objectius del projecte considero que els he assolit en general, tot i que el projecte ha acabat diferent del que tenia pensat, les meves expectatives no estaven basades en l'experiència, i ara que sí que se com és el procés i tot el que comporta la creació d'un videojoc, no puc més que sentir-me orgullós d'haver après tot el que he après i haver superat els diferents obstacles que se m'han presentat, aixentant-me cada cop que em feien caure, justament com el nostre personatge en el modo de joc Persecució.

Durant el procés de creació de videojoc, aquest ha anat canviant de rumb per circumstancies externes, ja que no era capaç d'arribar als meus objectius inicials amb els meus recursos, la falta de temps i de salut han estat un factor a tenir en compte, ja que durant el primer mes de projecte vaig emplear cada moment lliure que tenia en aquest projecte i, finalment, vaig acabar amb tendinitis als canells. Pero com fa el nostre personatge en el mode de joc de combat, quan no tenim suficient energia hem de saber esperar per recuperar-la i poder fer el nostre següent cop, sense fallar ni acabant derrotats per no saber-nos gestionar.

Conceptualització

Aquest projecte porta existint a la meva ment mesos abans de començar el semestre, pel que la seva conceptualització ha sigut realment el que més temps ha portat de tot el projecte, tot i que normalment no es té gaire en compte. Les idees s'han d'anar descartant o polint per poder obtenir un resultat bó, i això porta molt de temps, ja que estem creant una obra de zero. Per això pot arribar punts en que sigui agobiant, sobretot per algú perfeccionista, ja que la nostra ment no dona més de si molts cops.

Va començar amb la premisa de que el videojoc volia explotar el que jo considero el factor que el diferencia de la resta de mitjans comunicatius o d'art, l'empatia i immersió que pot crear.

Fent referència a la primera PAC d'aquest treball de fi de grau, en que vaig explicar aquest concepte amb claredat:

“Els videojocs són un mitjà d'art que, comparat a d'altres com el cine, la pintura o la escultura està molt poc explorat, pel que molta gent encara no el veu com un mitjà d'art. Ara bé, per defensar que els videojocs són art hem de poder comparar-los als altres mitjans al mateix nivell. Podem estar d'acord en que l'art és una creació humana que expressa un sentiment, un pensament, un moment, etc, tot i que cadascú tindrà la seva propia definició. Cada mitjà d'art té els seus punts forts, característiques de les que no disposen els altres mitjans, una forma de transmetre el que es vol especial, fent més efectiu el missatge segons l'art que es tria per fer-ho.

Els videojocs, el punt més fort que tenen és la interacció, ja que fan que l'usuari, el consumidor de l'art, prengui part en el que és el videojoc. Si bé no en tots els videojocs hi ha el mateix nivell d'interacció, aquesta interacció ajuda a crear empatia, ja que l'usuari mateix forma part de l'experiència de manera activa. Per tant, un dels punts forts dels videojocs és la capacitat de generar empatia per sobre dels altres mitjans, ja que les possibilitats són il·limitades.

Depèn del tipus de videojoc i de com conecta amb els usuaris, el que vol expressar i com de bé ho fa podem tenir diferents graus d'empatia. Dels diferents graus d'empatia es poden definir 4 grans tipus diferents:

-**Empatia 0**, directament et donen una jugabilitat amb el fi d'entretenir sense més. Podriem definir això com art? Si només volen causar entreteniment sense transmetre cap missatge ni emoció estariem parlant d'art? Paral·lelament, és una foto de familia art? Si bé es fa servir el mitjà de la fotografia, amb el qual es pot fer art, no ha de ser art tot el que es faci. Per tant, els videojocs que només donen una jugabilitat entretinguda podrien no considerar-se art, depèn del punt de vista i de la definició que li volem donar. Un exemple d'aquests tipus de joc són els coneguts típics jocs de mòvil, com pot ser clash royale.



-Empatía inmersiva, són els videojocs en que el protagonista, qui el juga, és directament una càmera, sense cap carisma ni intervenció en els fets que s'expliquen. Un exemple d'això són els cada cop més coneguts walking simulator, que permeten a l'usuari explorar l'entorn i conèixer la història per si mateixos sense cap paper rellevant en ell. Un exemple d'aquests tipus de joc és Dark Souls: tot i que l'usuari té una tasca i una meta que aconseguir, el videojoc tracta d'explorar el que ara són unes runes del que va ser un país viu fa temps, descobrint el que va passar en cada localització sense poder intervenir, ja que el que ha passat no es pot canviar. El jugador és un observador passiu i la seva meta és més aviat una excusa per tal de donar una raó d'aquesta exploració. Aquesta jugabilitat es centra en la història i en que el jugador pugui entendre què passa i perquè passa.



-Empatía personal, els videojocs en que el jugador té un avatar propi en el videojoc, amb el qual pot triar decisions, interactuar amb els altres personatges i entorn i tenir una forta participació en la història. En aquests, l'usuari pot interaccionar i decidir com si ell mateix estigués vivint la història, fent que pugui empatitzar molt més, donant més realisme al que passa i visquent realment els aconteixements d'una manera molt pròxima. Un exemple clar d'aquest tipus de videojoc són els jocs de rol, posant d'exemple Dragon Age: Origins. Pots sentir molta empatia per la història i els personatges, ja que les decisions que prens afecten a tothom i causen uns comportaments o uns altres. Realment l'usuari s'esforça per fer les coses com ell vol i entendre el que

l'envolta per aconseguir-ho. En contraposició a la empatia inmersiva, en aquest cas l'usuari té un paper important que pot canviar moltes coses però degut al protagonisme es perd una mica la història, ja que l'usuari pot canviar-la i fer que el que es vol transmetre sigui diferent per cada jugador. Tot i això Dragon Age: Origins ho fa d'una manera molt coherent i transmetent el missatge desde diferents punts de vista.



-Empatía de canvi de pell, en els videojocs en que l'usuari controla a un personatge amb la seva pròpia personalitat i completament extern de la persona que el juga. Si bé en molts casos el jugador pren decisions, normalment les decisions solen tenir una coherència dintre del personatge, no podent canviar com és ell segons les accions del jugador.

Un exemple d'això és The Witcher III, en que el jugador ha de controlar a Geralt de Rivia, un personatge amb els seus propis llaços i el seu caràcter. Encara que el jugador tria moltes decisions, no ens fa canviar la manera en que veiem a Geralt, són decisions que ell mateix podria prendre en qualsevol dels casos, segons l'humor d'aquell dia. El punt fort d'aquesta empatia és que l'usuari literalment es posa en la pell d'una altre persona, visquent el que l'altre viu i poguent entendre un altre punt de vista extern, arribant a compenetrar-se amb el personatge de tal forma que sap com actuarà i perquè.



Si bé hi han jocs que no estan clarament definits per un d'aquests punts, cada joc és únic i es defineix a si mateix sota els seus termes, pel que mai estaran clarament categoritzats.”

Si bé la meua concepció del que es pot definir art ha canviat desde que vaig començar el projecte, penso que definir el que és art o no és subjectiu, ja que depén de si transmet alguna sensació, emoció o missatge i d'aquesta forma s'expressa. Podriem dir que una foto d'un lloc feta per qualsevol persona pot no ser art, però realment mitjançant aquesta foto podem veure que aquella persona va estar en aquell lloc, podem pensar que li va portar a estar allà i què va sentir per fer aquella fotografia.

A banda del que es considera art, la idea inicial del projecte era combinar diferents tipus d'empatia, el que no ha estat possible degut a la falta de la narrativa. Tot i això, és *Sendas pasadas* un joc d'empatía 0?

Com bé vaig indicar al final, no podem dir sempre a on està un videojoc en aquesta escala, ja que molts cops pot combinar diferents tipus alhora. Per això mateix tampoc consideraria *Sendas pasadas* un joc sense empatía, ja que fa servir l'empatia de forma diferent. Podem dir que aquest videojoc és una base solida a on podem construir una narrativa immersiva i empatica, ja que el gameplay està enfocad principalment a que el jugador es senti en la pell del personatge, que senti que està lluitant o perseguint i que visqui una experiència realista.

Sendas pasadas en un principi de concepció era un videojoc de l'actualitat en el que un nen tenia el poder de viure la vida d'altres persones, però el jugador hauria d'empatitzar amb aquests personatges i fer el que ells farien, indicant-ho amb una barra d'empatia.

Però no vaig trigar en adonar-me que aquesta barra treia al jugador de la immersió i feia que s'adonés en tot moment que estava jugant un videojoc i que estava empatitzant, no era un procés fluït i natural.

Finalment, vaig decidir que volia fer un videojoc sobre l'epoca medieval, ja que l'ambientació és la que més m'agrada i amb la que més agust treballaria, tot i

que va sorgir el problema d'haver de dibuixar bosc, un gran handicap per mi, que els cops que ho havia intentat se m'havia donat malament.

Amb aquestes dues idees clares després venia el procés de disseny, amb el que m'estaria molt de temps i descartaria diferents idees per la seva impracticitat o imperfecció.

Disseny

Tinguent ja un setting adequat, establert anteriorment, les possibilitats es reduïen, tot i que amb tots els videojocs que he arribat a jugar tenia moltes possibilitats en ment, tant de millorades d'altres jocs com de noves que mai havia vist però que m'hagués agradat veure.

Sendas pasadas tenia la idea de ser un joc empàtic, i per això una premisa bàsica era que tot el videojoc havia de ser realista, en quant a moviments, funcionalitats, estètica, havia de ser el més semblant a la realitat, sempre amb els límits del dibuix vectorial amb 2D que és el que sempre m'ha agradat més i més m'he dedicat a practicar. Per això, havia d'eliminar el màxim possible els elements en pantalla que no tindria el personatge en aquella situació, la idea principal és que el jugador visqui la història amb les mateixes circumstàncies en les que ell ho faria si estigués en la mateixa situació. Per tant, tots els elements en pantalla havien de ser els que veuria el personatge, tot i que per limitacions del format he hagut d'afegir alguns elements, com pot ser el menú. Tot i això, aquests elements també estan centrats en el que veuria el personatge i, per tant, treient la sensació d'estar jugant a un videojoc constantment.

El disseny del joc va ser gairebé tot al mateix temps, millorant algunes parts amb el temps i rectificant el romb però sempre seguint els criteris establerts al principi. Per poder explicar-ho més centralitzat i estructurat ho separaré en els diferents apartats de disseny que té el videojoc.

Icona d'aplicació

En el moment de dissenyar la icona per Sendas Pasadas volia fer un gràfic que mostres el que era sendas pasadas, però que alhora tingués les lletres S i P incorporades per ser les inicials del videojoc. Per tant, la S vaig decidir fer un camí, i la P va acabar seguint un arbre modificat, trets característics del videojoc i que veuriem durant tot aquest. Sobretot, no volia destacar les espases ja que no volia centrar-lo en això, no volia que el videojoc es veies només com un "joc d'espases", sino sobre un camí amb el seu paisatge i elements.

El menú

En aquest cas enfrontavem el repte més gran per part del realisme, el menú de joc, que només entrar al videojoc ja hi hagi un element que diu al jugador: "ei!

Que estàs jugant a un videojoc!" era preocupant. Per tant, la idea era implementar el menú com si fos part de la història o del ambient que es volia.

Per fer això, després de diferents opcions vaig triar un menú en que cada opció fos una espada, i mitjançant una cinemàtica inicial el personatge hauria d'agafar una espasa i aquest temps de menú seria interpretat com una indecisió del protagonista, mentre que anavem recorrent totes les opcions com faria algú en un comerç abans d'escollir què comprar.

Però també havia de tenir un element d'indicar a on estavem apuntant, ja que sino podria ser confós. Primerament vaig pensar en posar la mà com a cursor però no queda gaire bé i és més difícil de fer. Per tant, vaig optar per decidir que cada cop que el cursor passava per sobre d'un element del menú, aquest sortís brillant.

D'aquesta forma a més de quedar bé, el comportament seria més semblant al d'un menú, fent que fos més intuïtiu.

Per la tipografia vaig buscar un tipus de lletra que fos medieval però fàcilment llegible, pel que finalment vaig trobar la font de google Medieval Sharp, que s'adapta perfectament al que volia per aquest videojoc i la seva ambientació.

D'altra banda les lletres són en color tronja perquè em recorda l'època medieval i perquè el protagonista té la samarreta interior d'aquest color, fent que sigui un element destacable.

Narrativa

Com he mencionat en l'anterior punt, la idea era posar una cinemàtica només entrar al videojoc que ens introduís una mica en la història, en la que el protagonista hauria d'escollir una espasa. Per desgracia, com he remarcat a l'inici d'aquest document, no he pogut implementar la narrativa que volia, ja que és molt costós de fer, tant d'elements a dibuixar, com animar i diversos elements necessaris. Per tant, una decisió va ser deixar de banda aquesta història que tenia incorporada, tot i que aquesta decisió va anar prenent-se paulatinament fins desaparèixer, pel que el disseny de la narrativa i la història estava creat. La idea era que un jove mut tenia el poder de veure el passat de les persones en el moment que aquestes prenen una decisió important en l'actualitat, com podia ser matar algú, per tal de poder entendre el perquè d'aquesta decisió i, per tant, poder empatitzar. D'aquesta forma, el protagonista hauria d'emprendre un viatge a la capital, i s'aniria creuant amb diferents personatges imperfectes durant la seva travesia pel camí, coneixent els personatges i, donat el moment, el seu passat.

Per tant, tot el disseny de la jugabilitat estava feta per anar acompanyada d'una història, en la que el jugador es posaria en la pell del personatge que controla, sentint el que ell sentia i transmetent cada situació per tal d'endinsar al jugador en l'ambientació.

Personatges

Els personatges havien de tenir una estètica de l'època medieval, enfatitzant els seus atributs personals amb elements que expliquessin el seu passat.

Però no era tant fàcil de dissenyar, perquè s'havia de tenir en compte que les meves habilitats per dibuixar i animar són limitades, pel que no els podia fer gaire complexes ni amb unes robes amb massa detall, ja que es mourien molt en la jugabilitat i seria difícil de no trencar la sensació de realisme.

Finalment, després de dissenyar els personatges conceptualment només vaig poder arribar a dibuixar i animar en totes les seves formes al protagonista, el qual és neutre tant de gènere com d'aparença, per tal de poder crear la resta de personatges modificant el dibuix d'aquest per adaptar-los a les diferents personalitats. Tot i això, podem veure com el protagonista llueix robes llises i simples, ja que és un personatge humil, que no li agrada destacar gaire i passa desapercebut, tot i que al final tots els problemes li acaben passant a ell. Un element a destacar és el seu coll alt de la samarreta, el que està per ocultar el que posteriorment veuríem com el seu passat en el que va quedar mut.

A més, la història estava enfocada en un bosc per poder fer els personatges principals sense haver d'entretenir-se gaire en dibuixar personatges secundaris de fons, ja que els recursos eren limitats.

Escenari

Els escenaris serien diferents segons el lloc a on estariem, però tots compartien els arbres de fons per estar en el bosc, pel que havia de treballar seriosament en dibuixar i animar bé aquest fons, pensant en una paral·laxi per donar realisme. Per tant, vaig dissenyar el fons en diferents capes per tal de poder-lo moure a diferents velocitats mentre els personatges es movien en pantalla, tot i que havia d'estar difuminat per poder centrar-se en l'aspecte principal, els personatges.

Finalment, com que havia treballat tant en aquest escenari i el videojoc no tindria història, vaig decidir deixar-lo pels diferents mètodes de joc, ja que afegeixen un component estètic important que m'agradava tenir visible. Com que no és un element principal, tampoc es fa repetitiu realment.

Uns altres elements d'escenari importants són els obstacles del mode de joc persecució. Aquests obstacles havien de ser variats per no donar avorriment i tenir diferents tamanyes i actituds amb el jugador que descriuré més endavant. Per tant, vaig començar a pensar en diferents elements que es pot trobar una persona, que van variant segons l'escenari. Només vaig arribar a dibuixar els elements del camí i del bosc, que al final han estat posats la majoria en el mateix mètode de joc ja que només en seria un.

A banda de pensar quins elements ens podríem trobar en un camí, també vaig haver de pensar com podrien ser per ser fàcilment de saltar amb un salt normal, ja que la detecció i animació de col·lisió amb un element i posterior animació de tipus "parkour" em semblava massa complicada i avançada pels meus coneixements.

Finalment, com que no vaig trobar suficients elements i s'acabarien repetint força, vaig dibuixar versions diferents de cada element, pel que no es fes repetitiu i donés una sensació de novetat i també repte, ja que cada element té una forma diferent.

Música

Ja que el videojoc i la seva jugabilitat anaven a ser empàtics per tal de transmetre el màxim al jugador que el jugués, també havia de posar música que enfatitzés aquests aspectes que volia en les diferents jugabilitats.

Aquí va sorgir un problema bàsic de disseny, i és que jo no podia dissenyar la música amb els meus coneixements nuls sobre aquesta disciplina, però tampoc podia posar qualsevol cançó, ja que podria espatllar la sensació durant el joc.

Devant d'aquest panorama vaig començar a buscar cançons amb CC per poder-la fer servir sense cap problema.

Després de molta cerca i d'anar guardant diferents cançons que m'agradaven tot i que no encaixaven gaire en el que volia, vaig trobar un canal de youtube amb una música que em semblava professional i molt similar al que m'agradaria, ja que estava carregada d'emocions. El seu nom és Steffen Daum. Com que no m'acabava de creure que fos CC li vaig deixar un comentari al seu creador per tal de demanar-li permís explícit per poder explotar la mina que havia trobat. Ell, encantat, em va donar permís per fer-ho, pel que va començar el procés de tria entre totes les que en tenia.

Després d'anar canviant segons anava pensant en el disseny del videojoc vaig decidir que una cançó que donés ànims i ganes de jugar, que fos épica sobretot, hauria d'estar al menú, pel que vaig triar conqueror of the seas:

<https://www.youtube.com/watch?v=ZcokFZ0cwx8>

Pel videojoc de persecució es va complicar més, perquè havia de ser una música tensa però que s'adaptés a un trajecte de correr, amb un ritme adequat i, sobretot, remarcant que era important que el personatge no perdés per les greus conseqüències. Finalment em vaig decantar per Last Resistance, que comença més aviat lenta i augmenta el ritme, amb una percussió que encaixa bé amb l'animació de correr:

<https://www.youtube.com/watch?v=XFS9clwXZjE>

Per el mètode de joc de combat havia de triar una altra cançó que estigués carregada de sentiments i que encaixés amb el ritme ràpid del combat. Només escoltar la cançó Between Dragons and Humans vaig saber que hauria de fer servir aquesta, ja que donava un ritme perfecte amb la seva percussió i començava lent, introduint a l'acció:

<https://youtu.be/yDWqtAywi9A?list=PLtjfsINuRVLtxr01JQ-JjMpU7zKslmZy4>

Després d'haver fet servir aquestes cançons i haver-les escoltat tantes vegades durant la producció m'he acabat enganxant no només a aquestes sino a la resta de les seves cançons i finalment vaig decidir fer-li una donació, que ell ha refusat explicant que preferia veure el material amb el que s'ha fet la seva música. Treballar amb artistes amb una actitud tant amigable és molt còmode.

Aletorietat

Els dos mètodes de joc que no depenen d'un altre persona són completament aleatoris. En el mètode de joc de persecució, tant la distancia a la que s'allunya l'objectiu com els obstacles que ens trobem és aleatori (sota certs paràmetres). Això fa que la jugabilitat no es faci avorrida i que cada partida sigui única, pel que la rejugabilitat de Sendas pasadas és un factor a tenir en compte. Potser guanyar un combat contra la IA no porta gaire temps, però perfeccionar el combat i l'adaptabilitat i, sobretot, els combats amb altres persones, fan que poguem jugar-lo durant moltes hores.

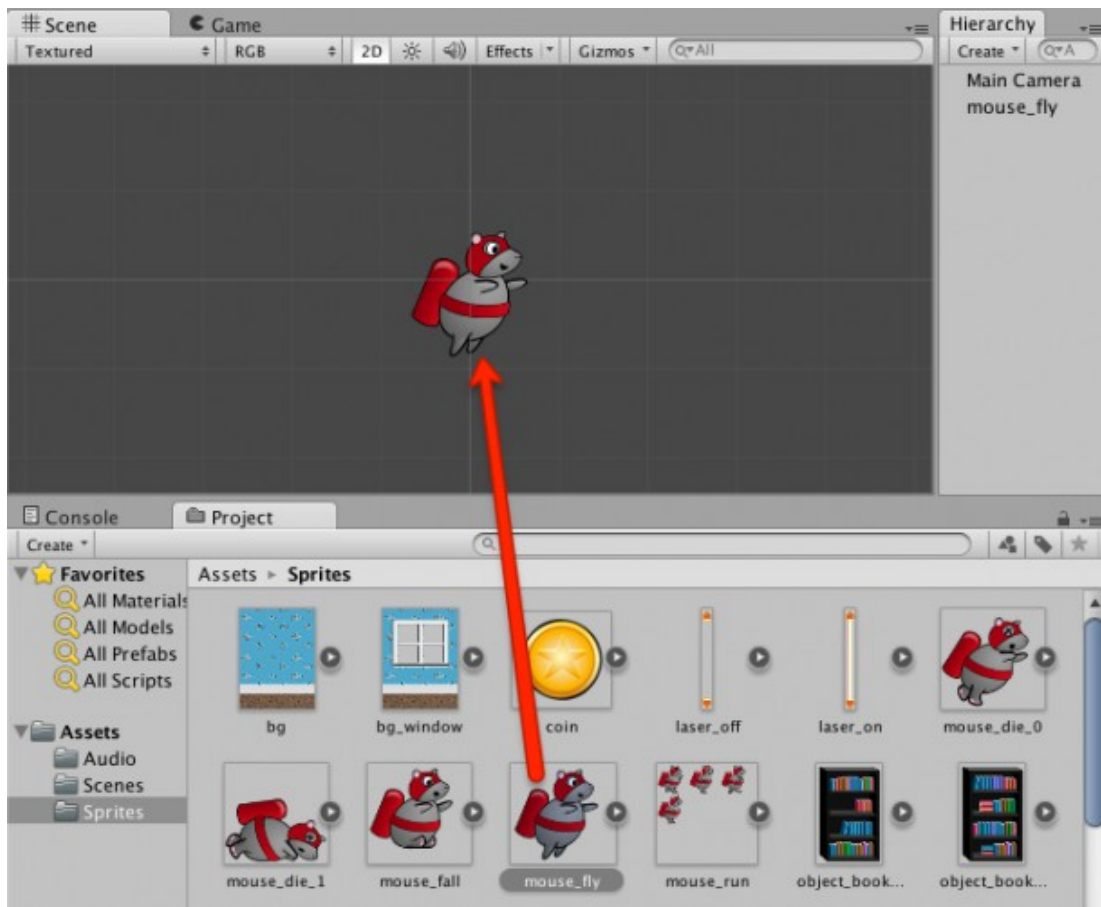
Persecució

Aquest mètode de joc va ser l'inicial amb el que em vaig desenvolupar en la creació del videojoc, pel que la seva implementació i disseny van anar lligades de la mà mentre anava avançant i aprenent.

La idea de fer un mode de joc en que s'havia de correr, sigués per perseguir o per fugir, ja estava dissenyada, però com seria i com funcionaria no, ja que no tenia gaire idea de com funcionava unity.

En veure un tutorial per internet en el que s'explicava pas per pas amb molt de detall i explicacions extenses com fer un videojoc de tipus jetpack, amb scroll lateral evitant obstacles, vaig veure clar que d'aquí podria fer servir molt material i, sobretot, aprendre a fer-ho.

<https://www.raywenderlich.com/69392/make-game-like-jetpack-joyride-unity-2d-part-1>



A l'implementació ja explicaré com va ser aquest procés d'aprenentatge. Ara bé, jo volia un moviment realista, amb diferents aspectes visuals dels que no disposava aquest tutorial, pel que vaig partir d'aquesta base però vaig modificar gairebé tot, poc codi es igual al que es fa servir en aquest tutorial.

Dissenyar d'aquesta manera, mentre s'implementa sobre una base ja establerta és molt fàcil realment, a banda de problemes d'implementació. Es tracta de seguir els mateixos passos i en veure com està implementat s'acudeix canviar un parell de coses per aquí... unes altres per allà... i al final s'acaven implementant noves funcionalitats, com és la de caure i aixecar-se.

Com que volia que hi hagués la mínima GUI però l'enemic estaria lluny de nosaltres havia de mostrar d'alguna forma el progrés, ja que el personatge que controlem està veient el seu objectiu, o al menys la seva silueta. Per això he dissenyat una GUI intuïtiva que ens mostra la distància amb l'enemic semblant a com ho veuria el personatge, amb una silueta que com més aprop més gran es fa i que, en cas de desaparèixer, l'hem perdut de vista definitivament. D'aquesta forma sembla gairebé un mirall retrovisor en els videojocs de conduir en primera persona, a on podem veure un element que no està disponible a la nostra visió però amb una fidelitat considerable.

Sobre les diferents tecles que fa servir, vaig intentar dissenyar un sistema innovador a on el jugador sentís realment el moviment de correr, saltar i aixecar-se, amb el mínim de tecles però el més fidel al moviment del personatge. Per això vaig pensar quan imitem a una persona amb la nostra mà, que cada dit es una cama i que abans de fer-lo saltar ens impolsem amb els dos dits, flexionant-los. D'aquesta forma, cada tecla seria una cama i, en premer les dues alhora, es realitzaria un salt.

Aquest disseny ha estat difícil de polir i té un problema principal i és que la gent no està acostumada a aquest tipus de controls, s'han de acostumar, el que pot portar una mica de pràctica.

Com que les animacions solen disposar de molts frames per fer l'animació més fluida, el moviment va ser difícil de crear i de fer el màxim de realista possible, perquè hem de pensar que no està corrent un atleta musculat, és una persona normal, que corre evitant obstacles amb nervis i cansanci.

Combat

El combat és el mateix disseny en els dos mètodes de joc que no són la persecució, el que canvia és que en un lluites contra la intel·ligència artificial i en l'altre amb una persona real. Aquest disseny ha estat el més elaborat i el que més temps s'ha trigat en fer, ja que és l'element principal i bàsic d'aquest videojoc.

La majoria dels videojocs tenen algun mode de joc o és el principal mode el combat, sigui del tipus que sigui. Després d'haver jugat a molts durant tota la meua vida he acabat aficionant-me als jocs de combats amb espases o altres armes semblants. Personalment no trobo tant disfrutable els combats amb poders màgics, amb trets o amb cops de puny, tot i que alguns videojocs que són de combats amb armes de fulla acaben seguint com aquests mencionats.

Fa molt de temps que sento a parlar que els videojocs de combats realment no són realistes i això m'ha arribat a fer pensar molt sobre què és cert i què no, ja que sempre fan els combats molt vistosos per tal de que el jugador es senti poderós però en l'epoca medieval no era així.

Segons vaig llegir, els combats d'espases realment es tracten més sobre els jocs de cames que no pas sobre les espases com a tal, ja que realment si estàs quiet i inmovil els combats d'espasa serien molt aburrits, fins que un es cansés. D'aquí neix la meua idea sobre el sistema de combat, intentant replicar un combat més realista, sempre amb els diferents elements dels que dispo i els meus recursos limitats. Per tant, en aquesta jugabilitat no es tracta de fer cops d'espasa ràpid o de qui es més fort, un cop d'espasa mata sigui a qui sigui.

Per tant, es tracta sobre la gestió dels recursos que es tenen i com es fan servir, cosa que sempre m'ha apassionat en els videojocs i són els que més m'han enganxat. Aquí no hem de fer servir una barra de vida, ja que em sembla un sistema molt comú però molt poc realista i immersiu, sobretot quan el gameplay és el mateix sense importar la vida que tens.

En el disseny del mode de combat de Sendas Pasadas s'ha de gestionar l'espai, amb la mobilitat de les cames, i la energia.

Com està a la taula del tutorial inicial reflexat, cada atac té un cost d'energia i guanya espai en respecte a l'enemic. A la mateixa vegada, cada atac té una possible defensa que gasta uns recursos d'espai o temps diferents per cada defensa. Per tant, no només hem de tenir cura per no perdre els nostres recursos sino que hem de sacrificar part dels nostres per poder fer que l'enemic també els perdi. Per exemple, si fem un tall inferior, gastarem molta energia, però l'enemic no el podrà evadir i li serà difícil de defensar, pel que perdrà molta energia, pero al mateix temps, com que l'atac no es gaire potent, no farà retrocedir gaire.

D'aquesta manera transcorre el combat per torns, amb un combat d'un cicle de 6 fases:

- El personatge prepara l'atac
- L'enemic es defensa
- El personatge executa l'atac
- L'enemic prepara l'atac
- El personatge es defensa
- L'enemic executa l'atac

Cada fase dura un màxim de 2 segons i un mínim de 0,75, per tal de poder realitzar les animacions correctament. Aquest breu temps de reacció fa que el jugador hagi d'improvitzar i que els seus atacs no siguin pensats al detall, sino que es valori la seva capacitat d'adaptació en el combat.

Amb un ritme ràpid però amb un ordre i seqüència, podem tenir la nostra estratègia sense que sigui aborrit i creant adrenalina pel temps de reacció.

També existeix la possibilitat de passar el torn quedant-se quiet, el que farà que es recuperi una mica d'energia i aquell torn no gastem recursos, però tampoc farem que l'enemic els gasti. A més, si fem un mateix moviment dos cops seguits, l'energia que fem servir es veura duplicada, pel cansament del mateix moviment tota la estona i per evitar abusos i nivellar la jugabilitat.

Els comandaments per moure el nostre personatge seran segons cap a on mogui la seva espasa, fent que primer hagi d'aixecar-la i, posteriorment, executi l'atac cap a baix.

Ara bé, per poder fer una jugabilitat tan profunda amb recursos sense tenir indicadors per pantalla d'aquest es necessita algun element que mostri de què disposem. Per això, a cada banda de la pantalla hi ha un arbre com a límit, en quant un toqui l'arbre, ensopegarà al no esperar-lo i perdrà. D'altra banda, per l'energia ens hem de fixar en l'animació d'espera del nostre personatge, com menys energia li quedi, més ràpid es moura per la seva accelerada respiració.

D'aquesta forma el jugador ha de tenir diverses coses en ment per fer la seva estratègia i ha d'adaptar-se a l'enemic, ja que segons el que faci l'enemic pot

fer que s'hagi de preocupar més pels propis recursos que per guanyar el combat.

Intel·ligència artificial

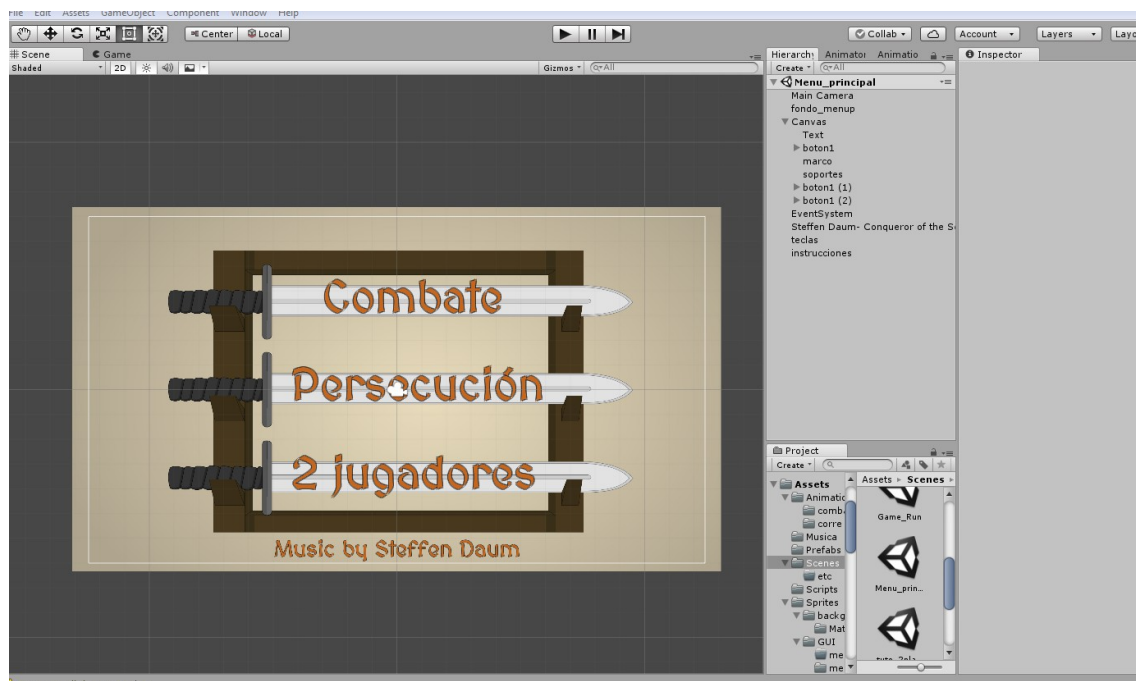
En el mode de joc de combat contra el sistema, he hagut de dissenyar una intel·ligència artificial bàsica per tal de que poguem lluitar contra un adversari i es faci una jugabilitat entretinguda. El nostre adversari disposarà d'uns elements igual que els nostres, però no podrà fallar cap defensa. Aquest factor no es gaire important tinguent en compte que defensar és realment fàcil, però per anivellar el videojoc, he fet que l'enemic no pugui executar el mateix atac dos cops seguits i que l'atac vertical superior no el pugui esquivar. A més, la seva estratègia és nula, ja que els seus atacs seran completament al atzar, el que fa que cada partida sigui única i ens hagim d'adaptar al màxim a les circumstancies que veiem, amb uns moviments imprevisibles més enllà de que no pot repetir-los.

Implementació

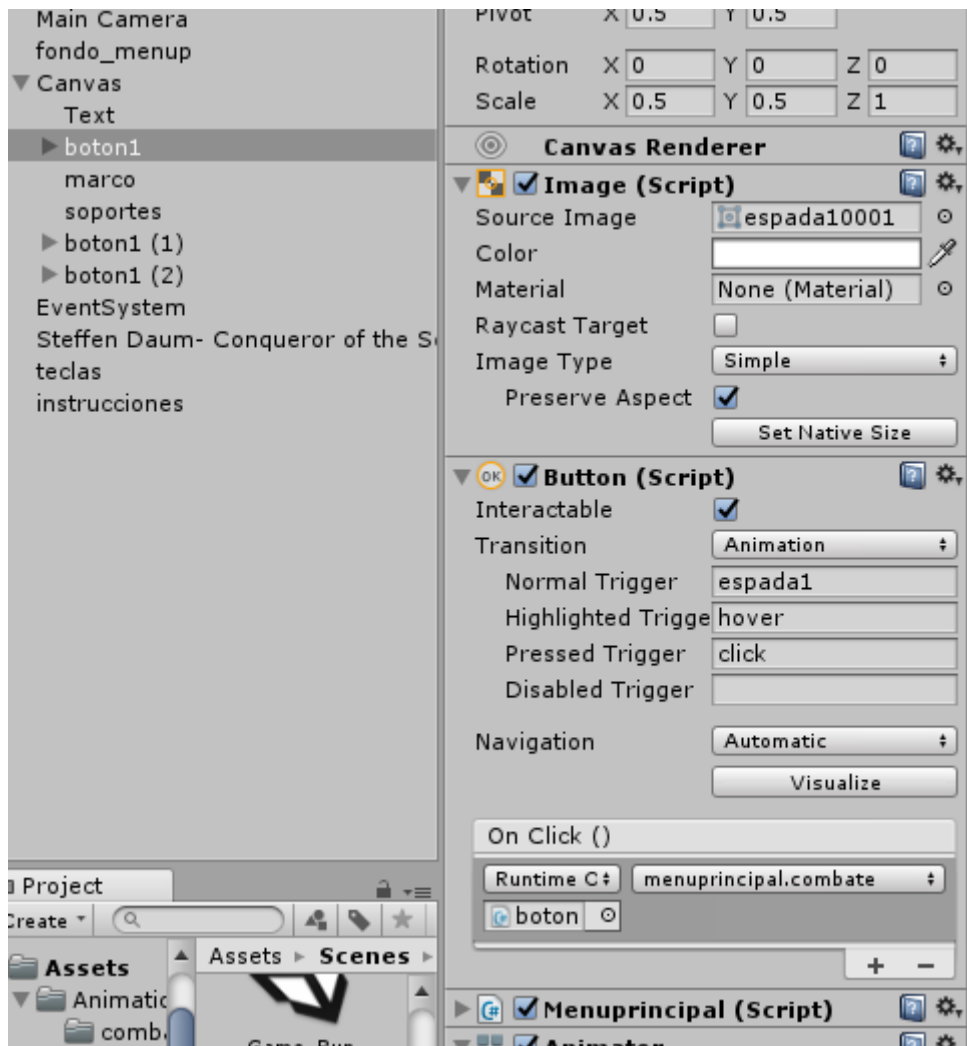
Per l'implementació d'aquest videojoc he fet servir Unity, amb c# com a codi de programació.

Cada escena té un script principal i segons la funcionalitat de l'escenari algun més.

Menú



El menú consta de 3 botons dins d'un canvas, amb el que cadascú té la seva imatge d'espasa i un text assignat al davant. D'aquesta forma, seleccionem que el metode de canvi d'estat sigui mitjançant una animació, en fer hover s'activarà l'animació del reflexe de llum.



Així mateix, mitjançant un script, indiquem que al fer click en cada botó s'executi una carrega d'escena diferent, depenent del que apremem. Aquest script també genera en ser cridat un botó a la part superior que ens servirà per sortir de l'aplicació, útil sobretot en pantalla completa.

```

}
void OnGUI(){
    Rect boton1 = new Rect (Screen.width * 0.87f, Screen.height * 0.01f, Screen.width * 0.1f, Screen.height * 0.05f);
    if (GUI.Button(boton1, "Salir")) {
        Application.Quit();
    }
}
}
public void combate () {
    SceneManager.LoadScene (1);
}
public void combate2() {
    SceneManager.LoadScene (2);
}
public void persecucion(){
    SceneManager.LoadScene (3);
}
public void persecucion2(){
    SceneManager.LoadScene (4);
}
public void jugadores(){
    SceneManager.LoadScene (5);
}
public void jugadores2(){
    SceneManager.LoadScene (6);
}
}

```

Tutorials

Per als tutorials he fet servir un sisetma similar al fet servir en el menú principal, amb el qual un botó transparent ocupa tota la pantalla, que en fer click, farà servir el mateix script que el menú principal per conduir a la següent escena. De fons tenim una imatge amb el tutorial explicatiu sobre com jugar-lo.



Paral·laxi

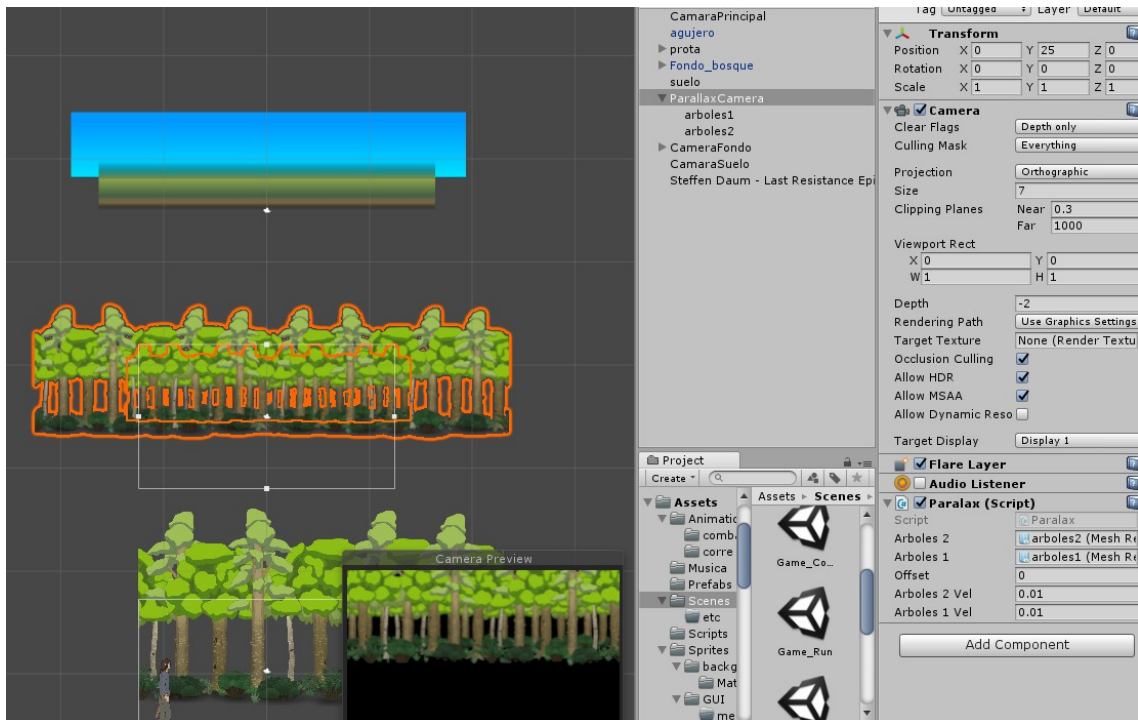
Per fer la paral·laxi hem de posar els elements que es mouran a diferents velocitats en un altre plà, amb una càmera diferent. Creem els objectes com a Quad i li assignem la imatge dels arbres com una textura, el que farà que vagi repetint-se si hem fet un sprite capicua.



Després creem un Script diferent que fixarem a la càmera de paralaxi, que rebrà un paràmetre del personatge, segons com es mogui, el que farà que es vagi moguent la textura pero no l'objecte i com que la càmera de paral·laxi estarà quieta, donarà igual com es mogui el player.

```
7   public Renderer arboles2;
8   public Renderer arboles1;
9
10  public float offset;
11
12  public float arboles2Vel = 0.01f;
13  public float arboles1Vel = 0.01f;
14
15  // Use this for initialization
16  void Start () {
17
18  }
19
20  // Update is called once per frame
21  void Update () {
22      float arboles2Offset = offset * arboles2Vel;
23      float arboles1Offset = offset * arboles1Vel;
24
25      arboles2.material.mainTextureOffset = new Vector2 (arboles2Offset, 0);
26      arboles1.material.mainTextureOffset = new Vector2 (arboles1Offset, 0);
27  }
28 }
```

A més, haurem de crear una altre capa amb una nova càmera per el fons dels arbres estàtics, ja que al ser scripts no es quedaran al darrera de tot i, mitjançant la profunditat de les càmeres haurem de determinar quina va davant de l'altre.



Per fer que el terra i els arbres del davant vagin també moguent-se sense parar amb cada moviment no he pogut fer-ho mitjançant el sistema de paral·laxi, ja que s'han de moure al mateix cop que el player i les capes no quedaven ordenades correctament si no creava una nova capa de nou per el terra.

Per fer que es vagin generant, copiant al lloc i esborrant els anteriors scripts ho fem mitjançant un script, que detectarà si fa falta generar un de nou o si ja ha sortit de pantalla i s'ha d'esborrar.

Amb aquesta funció comprovem si hem de generar o esborrar un nou script:

```

void compruebaCamino() {
    //creamos la array de los caminos que hay que borrar
    List<GameObject> caminoBorrar = new List<GameObject>();
    //indicamos que crearemos el camino para luego desactivarlo si es necesario
    bool addCamino = true;
    //recojemos la posición X del player
    float playerX = transform.position.x;
    //indicamos la distancia en la que borraremos el camino
    float borraCaminoX = playerX - distPantalla;
    //indicamos la distancia en la que creamos el camino
    float creaCaminoX = playerX + distPantalla;
    //creamos la variable que indicará la distancia del ultimo camino
    float final = 0;
    //por cada camino que esté creado
    foreach (var camino in bosqueActual) {
        //recojemos el tamaño del objeto arboles para indicarselo al sprite
        Vector2 sprite_size = camino.transform.Find ("camino_suelo").gameObject.GetComponent<SpriteRenderer> ().sprite.rect.size;
        //el ancho total del camino depende del ancho de los arboles
        float ancho = sprite_size.x/87.55f;
        //indicamos la distancia a la que empieza
        float caminoEmpieza = camino.transform.position.x + (ancho*0.5f);
        //sumandole el ancho indicamos la distancia a la que acaba
        float caminoAcaba = caminoEmpieza + ancho;
        //si la distancia a la que empieza es mayor de la que pedimos para crear un camino indicamos que no hace falta crear más
        if (caminoEmpieza > creaCaminoX) {
            addCamino = false;
        }
        //si el camino está fuera del rango permitido procedemos a borrarlo
        if (caminoAcaba < borraCaminoX) {
            caminoBorrar.Add (camino);
        }
    }
    //con esto indicamos que si el final del camino actual es más grande que el anterior, guardemos esta distancia
    final = Mathf.Max (final, caminoAcaba);
}

```


Finalment, afegim el camí on calgui o esborrem part del camí que ja no sigui visible.

```
}
//por cada camino en la array de borrar lo borramos
foreach (var camino in caminoBorrar) {
    bosqueActual.Remove (camino);
    Destroy (camino);
}
//si no hay suficiente camino, creamos uno nuevo, indicandole la posición donde debe crearse
if (addCamino == true) {
    creaCamino (final);
}

}

void creaCamino (float final) {
    //Llamamos al prefab del camino y lo creamos
    GameObject camino = (GameObject)Instantiate (bosquePrefab [0]);
    //Le indicamos que cumpla con el ancho del suelo
    float ancho = camino.transform.Find("camino_suelo").localScale.x;
    //Le decimos donde estará su centro, sumando la posición del anterior camino y la mitad del actual
    float centro = final + ancho * 0.5f;
    //Le indicamos la posición donde debe estar
    camino.transform.position = new Vector3 (centro, 0, 0);
    //añadimos el camino a la array de caminos existentes
    bosqueActual.Add (camino);
}
}
```

Amb la funció de crear el camí, agafa la posició a on ha de crear-lo i crea una instància del prefab generat a partir dels dos sprites.

Càmera

La càmera ha de seguir al jugador tota la estona, alhora que aquest es mou, pel que ha de rebre el paràmetre mitjançant script sobre el moviment del jugador i amb aquest moure la càmera al mateix temps.

```
public GameObject objetivo;
private float distanciaX;
// Use this for initialization
void Start () {
    distanciaX = transform.position.x - objetivo.transform.position.x;
}

// Update is called once per frame
void Update () {
    float objetivoX = objetivo.transform.position.x;
    Vector3 posicion = transform.position;
    posicion.x = objetivoX + distanciaX;
    transform.position = posicion;
}
}
```

D'aquesta forma mantenim el personatge en el mateix lloc a on l'hem col·locat al inici i al moure's es mou en conjunt.

Persecució

El mètode de joc de la persecució disposa de diferents implementacions. Per un costat hem de generar els obstacles, que haurem convertit en prefabs anteriorment per poder-los duplicar i instanciar a l'escenari infinitament. Com feiem amb la implementació de crear i esborrar el camí farem el mateix amb els obstacles, amb la diferència de que aquests es generaran aleatoriament entre una array d'elements diferents.

```
void compruebaObstaculos (){
    List<GameObject> obstaculosBorrar = new List<GameObject>();
    float playerX = transform.position.x;
    float borraObstaculoX = playerX - distPantalla;
    float creaObstaculoX = playerX + distPantalla;
    float ultPosX = 0;

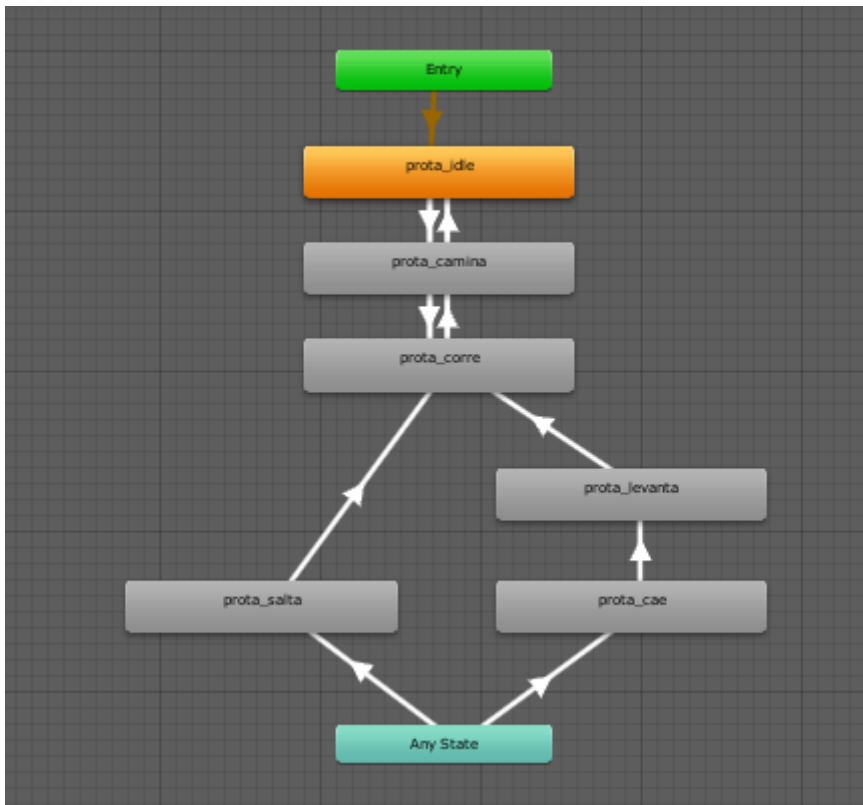
    foreach (var obstaculo in obstaculosActuales) {
        float obstaculoX = obstaculo.transform.position.x;
        ultPosX = Mathf.Max (ultPosX, obstaculoX);
        if (obstaculoX < borraObstaculoX) {
            obstaculosBorrar.Add (obstaculo);
        }
    }
    foreach (var obstaculo in obstaculosBorrar) {
        obstaculosActuales.Remove (obstaculo);
        Destroy (obstaculo);
    }
    if (ultPosX < creaObstaculoX) {
        creaObstaculo (ultPosX);
    }
}

void creaObstaculo (float ultPosX) {
    int random = Random.Range (0, obstaculosPosibles.Length);
    GameObject obstaculo = (GameObject)Instantiate (obstaculosPosibles [random]);
    float posX = ultPosX + Random.Range (obstaculosMinDist, obstaculosMaxDist);
    float posY = obstaculo.transform.position.y;
    obstaculo.transform.position = new Vector3 (posX, posY, 0);
    obstaculosActuales.Add (obstaculo);
}
```

Com feiem abans, comprova si fa falta generar un objecte nou per la seva posició, i el genera entre un rang indicat perquè no estiguin separats tots a la mateixa distància.

Així mateix l'esborra de l'array d'elements existents i destrueix la instància.

Per al jugador farem servir unes animacions amb unes variables booleanes que anirem canviant mitjançant codi, comprovant quan faci falta, segons la velocitat o si ha caigut l'usuari.



```

void anima(){
  if (velocidad.x == 0) {
    animator.SetBool("camina", false);
    animator.SetBool("corre", false);
  } else if (velocidad.x < 7 && velocidad.x > 0) {
    animator.SetBool("camina", true);
    animator.SetBool("corre", false);
  } else if (velocidad.x >= 10) {
    animator.SetBool("camina", false);
    animator.SetBool("corre", true);
  }
}
}

```

Cada frame farem una comprovació de diversos aspectes:

Per una banda la velocitat a la que va, fent que no pugui sobrepassar uns límits.

Per altra banda comproba la posició per, en cas d'estar al terra o estar caient, executar unes funcionalitats o unes altres.

Fem que l'usuari no pugui tenir velocitat negativa per que no retrocedeixi.

En cas de que estigui saltant, comproba si un element que té fixat als seus peus toca l'element del terra, per tal de parar el salt.

Indiquem la velocitat X si estem saltant per tal de fer un salt amb potencia i poder superar els obstacles.

```

void FixedUpdate () {
    //reproducimos la animación según la situación
    anima ();
    //Guardamos la velocidad en una variable alternativa para poder modificar sus valores
    velocidad = rigidbody.velocity;
    //comprobamos si la velocidad del personaje es mayor a la máxima y no está saltando para así limitarla
    if (velocidad.x > maxVel && !animator.GetBool ("saltando")) {
        velocidad = new Vector2 (maxVel, velocidad.y);
        rigidbody.velocity = velocidad;
    }
    if (animator.GetBool ("saltando")) {
        velocidad = new Vector2 (25, velocidad.y);
        rigidbody.velocity = velocidad;
    }
    //Añadimos velocidad hacia arriba que no se nota normalmente pero que hace el salto más largo
    velocidad = new Vector2( velocidad.x, (velocidad.y+0.5f));
    //con esto impedimos que el personaje retroceda hacia atrás si hay algún imprevisto
    if (velocidad.x < 0) {
        velocidad = new Vector2( 0, velocidad.y);
    }
    //efectuamos los cambios indicados en la velocidad
    rigidbody.velocity = velocidad;
    //si no está en el aire y no se ha quitado el control del personaje se ejecuta la función para comprobar si
    if (!animator.GetBool ("saltando") && !quitaControl) {
        paso ();
    } else if(animator.GetBool ("saltando")) {
        //comprueba si está saltando cuando aterriza en el suelo
        enTierra = Physics2D.OverlapCircle (compruebaSuelo.position, 0.1f, suelo);
        if (enTierra){
            animator.SetBool ("saltando", false);
        }
    }

    //si está con colisión está reproduciendo la animación de caer
    if (colision && quitaControl) {
        if (cooldown > 50) {
            colision = false;
            quitaControl = false;
            levanta = true;
            cooldown = 0;
        } else {
            cooldown = cooldown + 1;
        }
    }
}

```

Cada frame també executa la següent funció, que comproba si l'usuari prem algun botó i, en cas de fer-ho, executa el pas o el salt segons correspongui. Aquesta funció està duplicada per tal de que, depenent de quin pas hagi fet l'anterior (dret o esquerra), només pugui fer el següent pas amb l'altre botó. També, comproba en cas d'estar al terra si es premen els dos alhora per tal d'aixecar-se.

```

void paso () {
    //vamos intercambiando entre derecha e izquierda
    if (dirPaso == true) {
        //comprobamos si presionamos el botón del paso actual y el otro
        bool stepDone = Input.GetButton("Right") || Input.GetButton ("Right2");
        bool jumpDone = Input.GetButton("Left") || Input.GetButton ("Left2");
        //si apretamos los dos a la vez
        if (stepDone & jumpDone) {
            //si hemos apretado los dos durante 3 iteraciones saltamos, sino añadimos 1 iteración al contador
            if (salto < 3) {
                salto = salto + 1;
            } else if (levanta) {
                //animación de Levantarse
                animator.SetBool("cae", false);
                quitaControl = true;
            } //solo saltamos si hace 40 iteraciones o más desde la ultima vez que saltamos
            } else if (cooldown > 40) {
                //reseteamos las dos variables
                salto = 0;
                cooldown = 0;
                //añadimos fuerza vertical y horizontal para simular el salto
                fuerza = new Vector2 (1700, fuerzaSalto);
                //la aplicamos al cuerpo
                rigidbody.AddForce(fuerza);
                animator.SetBool ("saltando", true);
            }
        }
        //si solo apretamos la tecla de dar el paso y no está caído
        else if (stepDone & !levanta) {
            //indicamos la velocidad para sumar velocidad del paso a la velocidad actual
            velocidad = new Vector2( velocidad.x + pasoVel, 0);
            //aplicamos la velocidad
            rigidbody.velocity = velocidad;
            //cambiamos de paso
            dirPaso = false;
            //reiniciamos el contador de salto para no saltar sin intención
            salto = 0;
        }
    }
    //vamos intercambiando entre derecha e izquierda
}

```

El salt té un cooldown per tal de que no es faci servir constantment i, per tant, abusar de la velocitat que dona el salt.

Amb el següent script calculem la distancia en la que es troba l'enemic, contant la distancia que hem recorregut i la que ha recorregut l'enemic de forma aleatoria entre un rang indicat.

```

}
//calculamos la distancia del objetivo
void calculaDistancia () {
    //calculamos la velocidad del enemigo aleatoriamente entre un rango de 5 arriba o abajo según lo indicado
    velEnemActual = Random.Range ((velEnem - 5), (velEnem + 5));
    //calculamos la distancia añadiendo la velocidad del enemigo y restando la propia
    distancia = distancia - velEnemActual + velocidad.x;
    //indicamos el porcentaje de distancia a la que se encuentra en porcentaje con la distancia máxima
    indicadorDesps = distancia / distanciaFin;
    //indicamos el tamaño que debe tener el indicador, dividiendo por el multiplicador anterior y multiplicando por el actual
    tamañoGUI = tamañoGUI / indicadorAntes * indicadorDesps;
    //indicamos que el tamaño posterior pasa a ser el anterior para la siguiente vez
    indicadorAntes = indicadorDesps;
    //cambiar tamaño del gui
}

```

Amb això modificarem el tamany del objecte de la GUI que ens indica la distància de l'enemic, que el pintarà en pantalla amb el següent codi:

```

public void tamañoIcon(){
    //Rect GUI_fondo_rect = new Rect (distPantalla -10, 20, 95, 95);
    Rect GUI_fondo_rect = new Rect (Screen.width-120f, 20f, 95, 95);
    float distX = 6/(tamañoGUI/2);
    float distY = 8/(tamañoGUI/2);
    GUI.DrawTexture (GUI_fondo_rect, GUI_fondo);
    if (tamañoGUI > 0.5f) {
        Rect GUI_icono_rect = new Rect (Screen.width-97 + distX*0.5f, 17 + distY, 35 * tamañoGUI, 70 * tamañoGUI);
        GUI.DrawTexture (GUI_icono_rect, GUI_icono);
    } else {
        //Rect GUI_icono_rect = new Rect (distPantalla+25, 50, 35 * tamañoGUI, 70 * tamañoGUI);
        Rect GUI_icono_rect = new Rect (Screen.width-85, 50, 35 * tamañoGUI, 70 * tamañoGUI);
        GUI.DrawTexture (GUI_icono_rect, GUI_icono);
    }
}
}

```

Amb una nova funció comprovem les colisions amb els objectes que es topa l'usuari i, segons la seva tag, el relentitzarà o el farà caure.

```

//comprobamos las colisiones
void OnTriggerEnter2D(Collider2D obstaculo){
    //si el obstaculo tiene la tag de cae
    if (obstaculo.gameObject.CompareTag("cae")) {
        //quitamos el control del personaje
        quitaControl = true;
        //reproducimos la animación de caída
        animator.SetBool("cae", true);
        colision = true;
        cooldown = 0;
    }
    //si el obstaculo tiene la tag de relentiza
    else if (obstaculo.gameObject.CompareTag("relentiza")) {
        //relentiza al personaje cada iteración que pasa en contacto
        velocidad = new Vector2(2, 0);
        //aplicamos dicha velocidad
        rigidbody.velocity = velocidad;
    }
}
}

```

Finalment, en cas d'haver perdut o guanyat activarem el menú de reinici, que generarà un missatge indicat segons el resultat i dos botons, un per reiniciar el nivell i un altre per retornar al menú.

```

: //comprobamos si hemos perdido para que aparezca el menú de reinicio
: void menuReinicio(string mensaje) {
:     //indicamos que hemos perdido para quitar el control del personaje
:     quitaControl = true;
:     //añadimos el botón de reinicio en pantalla
:     Rect boton1 = new Rect (Screen.width * 0.35f, Screen.height * 0.35f, Screen.width * 0.3f, Screen.height * 0.1f);
:     if (GUI.Button(boton1, "Reinicia")) {
:         //en apretar al botón reiniciamos el nivel
:         SceneManager.LoadScene (4);
:     }
:     //añadimos el botón de volver al menú en pantalla
:     Rect boton2 = new Rect (Screen.width * 0.35f, Screen.height * 0.55f, Screen.width * 0.3f, Screen.height * 0.1f);
:     if (GUI.Button(boton2, "Vuelve al menú")) {
:         //en apretar al botón volvemos al menú
:         SceneManager.LoadScene (0);
:     }
:     GUI.skin.font = sharp;
:     GUI.contentColor = Color.black;
:     GUI.skin.label.fontSize = 50;
:     var centeredStyle = GUI.skin.GetStyle("Label");
:     centeredStyle.alignment = TextAnchor.UpperCenter;
:     GUI.Label (new Rect (Screen.width/2-500, Screen.height/4-25, 1000, 100), mensaje, centeredStyle);
:     var color = new Color (0.76f, 0.4f, 0.13f, 1);
:     GUI.contentColor = color;
:     GUI.skin.label.fontSize = 50;
:     GUI.Label (new Rect (Screen.width/2-502, Screen.height/4-26, 1000, 100), mensaje, centeredStyle);
: }
: //comprobamos las colisiones

```

Combat

El sistema de combat i de 2 jugadors és exactament el mateix, només que el de 2 jugadors repeteix el mateix codi canviant de tecles, mentre que el de combat fa una intel·ligència artificial.

En aquest mètode farem servir 6 passos diferents, entre els quals crearem uns temporitzadors que detindran l'acció per tal de que les animacions es completin correctament.

En el primer pas detectem el moviment de l'usuari i apliquem un descompte d'energia, activant el temporitzador i el següent pas.

```

void pasol (){
    animator.SetBool ("Ataca", true);
    animator.SetBool ("Defiende", false);
    //si el tiempo de respuesta es mayor que el temporizador aún no se ha cumplido el tiempo y puede efectuar un movimiento
    if (tiempoRespuesta > temporizadorRespuesta) {
        //añadimos más tiempo al temporizador
        temporizadorRespuesta = temporizadorRespuesta + Time.deltaTime;

        //si el primer movimiento es hacia la izquierda
        if (Input.GetButton ("Left") || Input.GetButton ("Left2")) {
            animator.SetBool ("Izquierda", true);
            //guardamos el movimiento hacia la izquierda
            movimientoPlayer = "left";
            //indicamos el tiempo que va a estar sin poder realizar ninguna acción para no interrumpir la animación ni el ritmo
            tiempo = 0.75f;
            temporizador = 0;
            //se indica el tiempo que tiene el player para realizar su siguiente movimiento
            tiempoRespuesta = 2;
            //se renueva el marcador para tener un tiempo independiente
            temporizadorRespuesta = 0;
            estado = 2;
            energiaPlayer = energiaPlayer - 5;
        }
        //si el primer movimiento es hacia la izquierda
        if (Input.GetButton ("Up") || Input.GetButton ("Up2")) {

```

En el segon pas, l'enemic prepara una defensa depenent de l'atac de l'usuari.

```
void paso2 (){
    switch (movimientoPlayer) {
        case "left":
            evade = true;
            enemianimator.SetBool ("Izquierda", true);
            animator.SetBool ("Defiende", true);
            //indicamos el tiempo que va a estar sin poder realizar ninguna acción para no interrumpir la animación ni el ritmo
            tiempo = 0.75f;
            temporizador = 0;
            //se indica el tiempo que tiene el player para realizar su siguiente movimiento
            tiempoRespuesta = 2;
            //se renueva el marcador para tener un tiempo independiente
            temporizadorRespuesta = 0;
            estado = 3;
            fuerzaEnem = 1000;
            reacEnem = 0.25f;
            energiaEnem = energiaEnem - 3;
            break;
        " "
    }
}
```

Finalment, depenent de l'atac que s'hagi preparat en un principi, en apretar la tecla correcta posterior, podrem executar l'atac, restant-se la energia necessària i acrivant la funció per fer el canvi de posició.

```
void paso3 (){
    animator.SetBool ("Ataca", false);
    //si el tiempo de respuesta es mayor que el temporizador aún no se ha cumplido el tiempo y puede efectuar un movimiento
    if (tiempoRespuesta > temporizadorRespuesta) {
        //añadimos más tiempo al temporizador
        temporizadorRespuesta = temporizadorRespuesta + Time.deltaTime;
    }
    //según en la postura que esté el personaje aceptaremos un movimiento u otro para seguir la secuencia
    switch (movimientoPlayer) {
        case "left":
            //si está preparando hacia la izquierda esperamos a que realice el movimiento hacia la derecha
            if (Input.GetButton ("Right") || Input.GetButton ("Right2")) {
                animator.SetBool ("Derecha", true);
                animator.SetBool ("Izquierda", false);
                enemianimator.SetBool ("Ataca", true);
                //restamos la energía del movimiento al player
                energiaPlayer = energiaPlayer - 10;
                //si el último ataque que realizó era este mismo recibe una penalización de energía por repetir, ya que el personaje se cansa más
                if (ultimoAtaque == "left") {
                    energiaPlayer = energiaPlayer - 10;
                }
            }
            //tras realizar el ataque esperamos a que la animación sea realizada quitándole el control al player
            tiempo = 0.75f;
            temporizador = 0;
            //se indica el tiempo que tiene el player para realizar su siguiente movimiento
            tiempoRespuesta = 2;
            temporizadorRespuesta = 0;
            estado = 4;
            //guardamos el último ataque que ha realizado
            ultimoAtaque = "left";
            fuerzaPlayer = 1000;
            reacPlayer = 0.25f;
        }
    }
    break;
}
```

Depenent del temporitzador a la reacció que hagim posat i la força aplicada a cadascú, s'aplicarà una força en aquell temps a cada cos.


```

void movPlayer() {
    //añadimos fuerza que implica la colisión
    fuerza = new Vector2 (fuerzaPlayer, 0);
    //la aplicamos al cuerpo
    rigidbody.AddForce(fuerza);
    //aplicamos lo mismo en negativo para el enemigo
    fuerzaPlayer = 0;
    reacPlayer = 0;
}
void movEnem() {
    //añadimos fuerza que implica la colisión
    fuerza = new Vector2 (fuerzaEnem, 0);
    //la aplicamos al cuerpo
    enemrigidbody.AddForce(fuerza);
    //aplicamos lo mismo en negativo para el enemigo
    fuerzaEnem = 0;
    reacEnem = 0;
}

void calculaEnergDist(){
    animator.SetFloat ("Energia", 4-energiaPlayer/energiaPlayerMax*3);
    enemanimator.SetFloat ("Energia", 4-energiaEnem/energiaEnemMax*3);
    if (energiaPlayer <= 0 || transform.position.x <= -12) {
        pierde = true;
    }
    else if (energiaEnem <= 0 || enem.transform.position.x >= 10) {
        gana = true;
    }
}
}

```

A més, cada frame calcularem l'energia que queda a l'usuari i a l'enemic per tal de modificar la seva animació per fer que vagi més ràpida conforme té menys energia.

També comproba si s'ha quedat sense espai per fer que es perdi o es guanyi.

Conclusions

Sendas pasadas ha estat un projecte del que estic molt orgullós, el qual he ensenyat a tothom que coneix i els hi he ensenyat a jugar.

He pogut veure que no és un joc que agradi gaire al públic estàndard, ja que no és una jugabilitat a la que estiguin acostumats, no és el típic videojoc, i és el que realment volia fer amb aquest projecte. Pot tenir els seus defectes, però és original i he après molt. Per mi és el més important, ja que per fer un joc clònic,

una mica canviat, prefereixo no fer-ho més enllà del que pugui aprendre, ja que no considero crear un joc amb modificacions d'un altre "Creació", sino millora, adaptació, etc.

La paraula clau que no defineix aquest projecte és eficiència, no he aprofitat els recursos dels que disposava degut al meu desconeixement, així com la història que vaig crear, la narrativa, alguns elements gràfics, idees, etc en els que vaig invertir temps finalment no s'han fet servir, no he pogut enfocar tot el projecte al que ha estat al final. Possiblement, si hagués enfocat el projecte com ha acabat, hagués sortit un producte més complet i polit.

Però el que si que defineix aquest projecte és la perseverancia, l'adaptabilitat i, sobretot, l'aprenentatge. Si bé he dedicat temps en coses que no han servit, ha servit perquè aprenés. Potser el meu videojoc no té narrativa, però he après a crear-la i desenvolupar-la, i això és amb el que em quedo. Potser si no hagués volgut fer una narrativa al principi el gameplay hagués estat més disfrutable... però ara mateix no sabia crear una narrativa i enfocar un videojoc en aquest aspecte.

Ha tingut tants canvis i retalls per poder dedicar-me al que importava que he après molt sobre la gestió d'aquests recursos i com es fa, quins recursos gasta cada procediment i com es pot crear un plà adaptable segons els esdeveniments.

Sobretot em deixa amb un bon sabor el fet d'haver-me esforçat molt, sense importar els diferents obstacles amb els que m'he topat i, si bé no he pogut saltar-los i he caigut, he après a donar-los la volta, pel que ja no em tornarà a passar. I aquest punt és el que més important em sembla del treball de fi de grau, aprendre, i sense cometre errors no podem aprendre tant.

Per això mateix veig la meva obra imperfecta com una obra perfecte com a tutorial per aprendre per mi, i una mostra del meu progrés durant aquests 4 mesos de dur treball, poguent dir que he creat un videojoc original, fet completament per mi.

