

Treball Final de Grau

Estudi: Grau en Disseny i Desenvolupament de Videojocs

Títol: Desenvolupament d'un videojoc sobre l'Esclerosi Múltiple

Document: Memòria

Alumne: Rubén Martínez Comino

Tutor: Gustavo Patow

Departament: Informàtica, Matemàtica Aplicada i Estadística

Àrea: Llenguatges i Sistemes Informàtics

Convocatòria (mes/any): Setembre / 2020



PROJECTE EM

Desenvolupament d'un videojoc sobre l'Esclerosis
Múltiple

Grau en Disseny i Desenvolupament de Videojocs

Ruben Martínez Comino
rubenm101096@gmail.com

AGRAÏMENTS

A en Gustavo Patow, tutor del treball, per haver-me donat confiança i suport des que li vaig comentar la idea del projecte.

Als doctors Lluís Ramió i René Robles, incloent a l'infermer Miquel Merchan, per oferir-me el seu temps i el seu interès per poder desenvolupar aquest treball amb la fidelitat mèdica que necessitava.

A la M. Begonya Matesanz, pacient diagnosticada amb Esclerosi Múltiple, per donar-me l'oportunitat de fer-li una entrevista i poder conèixer la malaltia des d'un punt de vista personal.

A les persones properes a mi que sempre han cregut en aquest projecte i la meva capacitat per dur-lo a terme, esmentant a Júlia Bozzino, Maria Sanz i Anna Bodro; juntament amb els meus pares.

A la Rosaura Molina, viuda de Xavier Massó, mort per una malaltia degenerativa, qui em va inspirar per desenvolupar aquest projecte.

ÍNDEX

1	INTRODUCCIÓ.....	6
1.1	Motivacions.....	6
1.2	PROPÒSIT	7
1.3	OBJECTIUS	7
1.4	QUADRE D'AUTOVALORACIÓ.....	8
2	ESTUDI DE VIABILITAT	9
2.1	RECURSOS TÈCNICS	9
2.2	RECURSOS HUMANS	9
2.3	REQUISITS TECNOLÒGICS.....	9
2.4	COST ECONÒMIC.....	10
2.5	ESTUDI DE MERCAT	11
2.5.1	CERQUES REALITZADES	11
2.5.2	RESULTATS OBTINGUTS.....	12
2.5.3	MATRIU DE COMPETITIVITAT	15
2.5.4	CONCLUSIONS ESTUDI DE MERCAT	16
2.6	PÚBLIC OBJECTIU I PERFIL DE JUGADOR	16
2.7	CONCLUSIÓ DE LA VIABILITAT DEL PROJECTE	17
3	PLANIFICACIÓ	18
3.1	PLA DE TREBALL.....	18
3.1.1	ABAST DEL PROJECTE	19
3.1.2	TASQUES PLANIFICADES.....	20
3.1.3	CRONOGRAMA.....	21
4	MARC DE TREBALL I CONCEPTES PREVIS	22
4.1	REFERÈNCIES	22
4.1.1	HELLBLADE: SENUA'S SACRIFICE	22
4.1.2	CELESTE	24
4.2	MOTOR DE JOC: UNITY.....	25
4.3	PEGI	27
5	ESCLEROSIS MÚLTIPLE.....	28
5.1	INTRODUCCIÓ.....	28
5.2	SÍMPTOMES.....	28
5.3	EXPERIÈNCIA AMB METGES I PACIENTS.....	29
5.4	ESCLEROSIS MÚLTIPLE EN EL VIDEOJOC	30
6	DISSENY DEL VIDEOJOC.....	33
6.1	ESTIL	33

6.2	GÈNERE.....	33
6.2.1	CÀMERA.....	34
6.2.2	GRÀFICS.....	34
6.3	GAMEPLAY.....	35
6.3.1	DEFINICIÓ DE REPTES	35
6.3.2	JERARQUIA DE REPTES	36
6.3.3	DEFINICIÓ D'ACCIONS	37
6.4	MECÀNIQUES	37
6.4.1	RECURSOS.....	38
6.4.2	ENTITIES.....	39
6.4.3	DESCRIPCIÓ DE MECÀNIQUES.....	39
6.5	FLOWCHART	41
6.6	LEVEL DESIGN	41
6.6.1	ECONOMIA INTERNA.....	42
6.7	CANVIS REALITZATS DE DISSENY	43
6.8	HISTÒRIA	44
6.8.1	MÓN DEL JOC	44
6.9	INTERFÍCIE GRÀFICA.....	45
6.9.1	MENÚ PRINCIPAL	45
6.9.2	NIVELL.....	46
6.9.3	MENÚ OPCIONS.....	48
6.10	ART DEL JOC	48
6.10.1	ESCENARIS	48
6.10.2	PERSONATGES	50
6.10.3	ALTRES SPRITES	53
6.11	EFACTES DE SO	54
7	IMPLEMENTACIÓ I PROVES.....	55
7.1	ENTORN.....	55
7.2	PERSONATGE.....	59
7.2.1	ANIMACIONS	60
7.2.2	PLAYER CONTROLLER	61
7.2.3	SYMPTOMS.....	64
7.3	IMPLEMENTACIÓ CÀMERA.....	64
7.4	IMPLEMENTACIÓ DELS SÍMPTOMES.....	65
8	RESULTATS.....	70
9	CONCLUSIONS	74

10	TREBALL FUTUR.....	75
11	BIBLIOGRAFIA.....	76
12	ANNEXOS.....	77
13	MANUAL D'INSTAL·LACIÓ I D'USUARI.....	85
13.1	MANUAL D'INSTAL·LACIÓ	85
13.2	MANUAL D'USUARI	86

1 INTRODUCCIÓ

En el món dels videojocs predominen les històries d'aventures; viatges on el/la protagonista s'enfronta amb perills i adversitats fins a complir un objectiu. En aquestes històries, la narrativa i les mecàniques treballen per posar al jugador en una fantasia de poder, amb personatges jugables que solen tenir habilitats sobrehumanes. Però un dels àmbits menys explorats en els videojocs és la de les malalties. Hi ha molt poques històries interactives on el/la protagonista pateixi alguna patologia, tant física com mental, i que per tant el jugador se senti en una situació de desavantatge.

Un dels pocs i recents exemples de videojocs que tracten sobre una malaltia és *Hellblade: Senua's Sacrifice*¹, que tracta la història d'una noia celta, Senua, que pateix de psicosis i s'embarca en un viatge al regne mític de Helheim per recuperar l'ànima del seu amant, mort pels invasors vikings. Tant la narrativa, com les mecàniques, les actuacions, els efectes de so i el disseny de nivells giren en torn a aquesta malaltia mental per a que el jugador s'hi posi en la pell (i en el cap, en aquest cas).



Figura 1 [Captura del Hellblade](#)

En aquest projecte s'estudiarà la malaltia degenerativa de l'Esclerosis Múltiple per fer la creació del prototip d'un joc del gènere de plataformes 2D on la protagonista patirà alguns dels símptomes d'aquesta malaltia.

1.1 Motivacions

Per norma general, en un treball, recerca, estudi, etc., hi ha dos tipus de motivacions amagades al darrere: les personals i les necessitats professionals. Les personals solen estar més encarades en les ganes que té un mateix de treballar o investigar en un tema en concret. En el meu cas, la motivació neix de la pèrdua d'una persona propera a causa

¹Joc de Ninja Theory: <https://www.hellblade.com/>

d'una malaltia degenerativa (Esclerosis Lateral Amiotròfica) i l'interès per voler transmetre les dificultats que suposa viure amb una malaltia d'aquest tipus.

En la societat actual, hi ha moltes patologies o discapacitats que són molt reconegudes però alhora estan poc visibilitzades, pel qual existeix una empatia limitada amb les persones que pateixen aquestes dificultats, una empatia que usualment es troba en les persones més properes. Per tant, és interessant donar visibilitat a una malaltia que pateixen a diari unes 2.5M de persones en tot el món, i aconseguir que més gent es pugui posar a la seva pell.

1.2 PROPÒSIT

Vist la baixa quantitat de videojocs que tracten sobre una malaltia, el propòsit principal d'aquest projecte serà desenvolupar el prototip d'un videojoc que posi als jugadors en la pell d'una persona amb Esclerosis Múltiple (EM). Aquest prototip no serà un *Serious Game* o un joc pensat per a persones amb aquesta patologia, sinó un videojoc lúdic del gènere de Plataformes 2D destinat a persones que no pateixin EM i puguin empatitzar més amb els afectats.

1.3 OBJECTIUS

L'objectiu d'aquest Treball Final de Grau és desenvolupar el prototip d'un videojoc del gènere de plataformes 2D que, invertint el sistema de progressió típic del disseny de videojocs, permeti posar-se a la pell d'algú amb Esclerosis Múltiple a jugadors que no pateixin la malaltia, per promoure'n la visibilització. Per aconseguir-ho, es mostraran les mecàniques principals del joc (moviment en 2D, salt, etc.) així com la representació d'alguns símptomes d'una persona amb Esclerosis Múltiple. Per dur a terme aquest projecte es faran reunions amb neuròlegs especialitzats en la malaltia i entrevistes amb pacients diagnosticats per tal d'obtenir un assessorament mèdic i personal que permeti donar una experiència més fidel a la malaltia. El projecte està centrat en l'apartat tècnic de la proposta, però també es treballarà superficialment en l'art i l'apartat gràfic. El joc es programarà en Unity i els elements gràfics es realitzaran amb Photoshop. Les tasques a desenvolupar inclouen, però no es limiten a:

- Estudiar el desenvolupament i els símptomes de L'Esclerosis Múltiple.
- Implementar els controls bàsics d'un joc de plataformes 2D.
- Implementar una representació dels diferents símptomes de la malaltia.
- Dissenyar nivells adaptats a les mecàniques seleccionades
- Redondeig i finalització de la documentació desenvolupada al llarg del projecte.

1.4 QUADRE D'AUTOVALORACIÓ

Normalment un videojoc és creat per un equip multidisciplinari, però en aquest cas només una persona s'encarregarà de totes les diferents parts que l'engloben. A continuació es mostra un quadre (Taula 1) amb una valoració personal de quant d'esforç i valor es dedica a cada un d'aquests quatre elements

Estètica	5%
Narrativa	5%
Mecàniques	60%
Tecnologia	30%

Taula 1 Quadre d'autoavaluació

El bloc on es gastaran més recursos és clarament el de les mecàniques. La proposta seleccionada està pensada per a que, a través de les mecàniques, el jugador es pugui posar a la pell d'una persona amb Esclerosi Múltiple. Les entrevistes amb neuròlegs i pacients serviran per decidir quins símptomes se seleccionaran per representar en el joc i de quina forma s'implementaran. La tecnologia és el 2n bloc més important ja que es treballarà amb el motor Unity i com es programin els símptomes influenciarà molt en l'experiència final.

L'estètica i narrativa tindran un pes molt baix ja que només es farà el disseny i l'animació de la protagonista mentre que els sprites de l'escenari seran extrets d'internet, i la història només servirà com a excusa narrativa per donar-li sentit als nivells i objectius.

2 ESTUDI DE VIABILITAT

En aquest capítol es valorarà la viabilitat del projecte des de diferents aspectes compoden ser: els recursos tècnics, els recursos humans, els requisits tecnològics o el cost econòmic.

2.1 RECURSOS TÈCNICS

Aquest treball s'ha realitzat en un sol ordinador, el sistema operatiu del qual és el Windows 10 de 64 bits. Pel que fa als components, té les següents especificacions:

- Processador: Intel® Core™ i7-7700HQ
- Tarjeta gràfica: NVIDIA GeForce GTX 1050
- Memòria: 8GB RAM

2.2 RECURSOS HUMANS

Com la majoria de videojocs, aquest treball requereix quatre perfils de persona:

- **Dissenyador principal:** Serà l'encarregat de plantejar com seran els nivells, quines mecàniques es faran servir i com funcionaran. També tindrà la responsabilitat d'estudiar les eines que es necessitaran juntament amb la seva documentació. Finalment s'encarregarà d'elaborar la memòria del projecte.
- **Programador i tester:** Serà l'encarregat de definir com haurà de ser l'algorisme, quines funcionalitats i mètodes haurà de tenir i com s'haurà de dur a terme per aconseguir-les. També tindrà la responsabilitat de comprovar que el joc funciona correctament.
- **Artista principal i guionista:** Serà l'encarregat de dissenyar i produir l'art del joc, ja sigui dels escenaris, els personatges, les animacions, etc.
- **Cap del projecte:** Serà l'encarregat de coordinar el projecte seguint la metodologia de treball escollida, ajustar els terminis de les diferents etapes de desenvolupament i aconseguir un bon funcionament en el treball.

Com que aquest projecte ha estat realitzat per una mateixa persona li han correspost els perfils d'analista i programador. Pel que fa al perfil de cap de projecte, l'han dut a terme el tutor del projecte.

2.3 REQUISITS TECNOLÒGICS

Pel desenvolupament d'aquest projecte, només ha calgut l'ordinador descrit a l'apartat de 'Recursos tècnics'. S'ha utilitzat el motor de desenvolupament Unity, en la versió 2019.2.7f2 (64-bit). Dins d'aquest, s'ha programat amb l'entorn de Visual Studio utilitzant el llenguatge de programació C#.

2.4 COST ECONÒMIC

En aquesta secció definirem el cost econòmic dels recursos tècnics, dels recursos humans i de la tecnologia que hem explicat en els apartats anteriors.

En el cas dels recursos tècnics, com que hem treballat amb un ordinador personal ens estalviarem el cost de comprar-lo, tot i això haurem de tenir en compte l'amortització dels recursos utilitzats. Per calcular-la utilitzarem la següent fórmula:

$$Amortitzacio = \frac{mesosfeina * preurecurs}{36} = \frac{8 * 1050}{36} = 233.33€$$

Pel que fa al *software*, tots els programes utilitzats són gratuïts.

En el cas dels recursos humans hem vist que, a banda del cap del projecte, són necessaris altres perfils: el dissenyador, el programador i l'artista. En aquest cas, tots dos perfils han estat duts a terme per la mateixa persona, però si haguéssim de calcular el cost per independent, tindriem que:

- Sou del dissenyador: 20€/h
- Sou del programador: 10€/h
- Sou de l'artista: 10€/h

Feines a realitzar:

- Estudi de les eines.
- Estudi de la malaltia.
- Realització d'entrevistes i reunions amb neuròlegs especialitzats i pacients.
- Disseny dels scripts de representació de símptomes.
- Disseny de nivells
- Disseny d'escenaris i personatges
- Animacions del personatge.
- Documentació.

Per tant, el cost en funció de les tasques a realitzar és el següent:

Tasques	Perfil	Hores	Cost/Hora	Cost Total
Aprenentatge Unity	Programador	20	10	200
Aprenentatge C#	Programador	25	10	250
Estudi malaltia	Dissenyador	20	20	400
Entrevistes i reunions	Dissenyador	10	20	200
Implementació scripts	Programador	100	10	1.000
Disseny de nivells	Programador	80	10	800
Disseny d'escenaris i personatges	Artista	15	10	150
Animacions del personatge	Artista	15	10	150
Documentació	Dissenyador	150	20	3.000
Total Programador		325	10	3.250
Total Analista		180	20	3.600
Amortització				233,33
TOTAL				13.233,33€

Taula 2 Cost Econòmic

2.5 ESTUDI DE MERCAT

Abans d'arriscar-se a invertir en un videojoc, és necessari estudiar la competència i consultar el mercat. En aquest apartat, entre d'altres es mirarà de valorar diferents qüestions com veure per a quin públic es dissenyarà aquest producte i perquè aquest públic l'escollirà. Al final de tot, es farà una matriu de competitivitat per poder posicionar la proposta seleccionada. En resum, és una forma de saber la resposta que obtindrem del mercat davant del producte seleccionat.

2.5.1 CERQUES REALITZADES

S'entendran com videojocs similars al que es proposa desenvolupar, tots aquells que tenen com objectiu un o varis dels següents ítems:

- Joc del gènere de plataformes 2D
- Joc sobre alguna malaltia (tant física com mental)

Tenint en compte aquests conceptes, s'han utilitzat com a paraules clau per fer les cerques:

- 2D Platform
- Disease Game
- Multiple Esclerosis Game
- Mental Health Game
- Pathology Game

S'han utilitzat les paraules soles i fent combinacions entre elles. El cercador principal utilitzat ha estat *Google*. També s'ha fet cerca en bases de dades especialitzades en jocs seriosos, com és la pàginade *serious games classification*, i s'han buscat articles acadèmics a *Google Scholar*.

2.5.2 RESULTATS OBTINGUTS

Els resultats obtinguts s'han analitzat i s'han classificat en els següents videojocs:

- **Medulla:** un plataformes 2D on s'ensenya el funcionament i l'estructura del cervell de forma lúdica i interactiva.
 - Cercador: *Google*
 - Paraules Clau: *2d platform pathology game*
 - Resultats: *3.100.000*



Figura 2 Captura del Medulla

- **MINWii Project:** joc seriós de teràpia musical amb l'objectiu de reduir alguns dels símptomes de pacients d'Alzheimer i demència.
 - Cercador: *Google Scholar*
 - Paraules Clau: *mental health video game*
 - Resultats: *510.000*

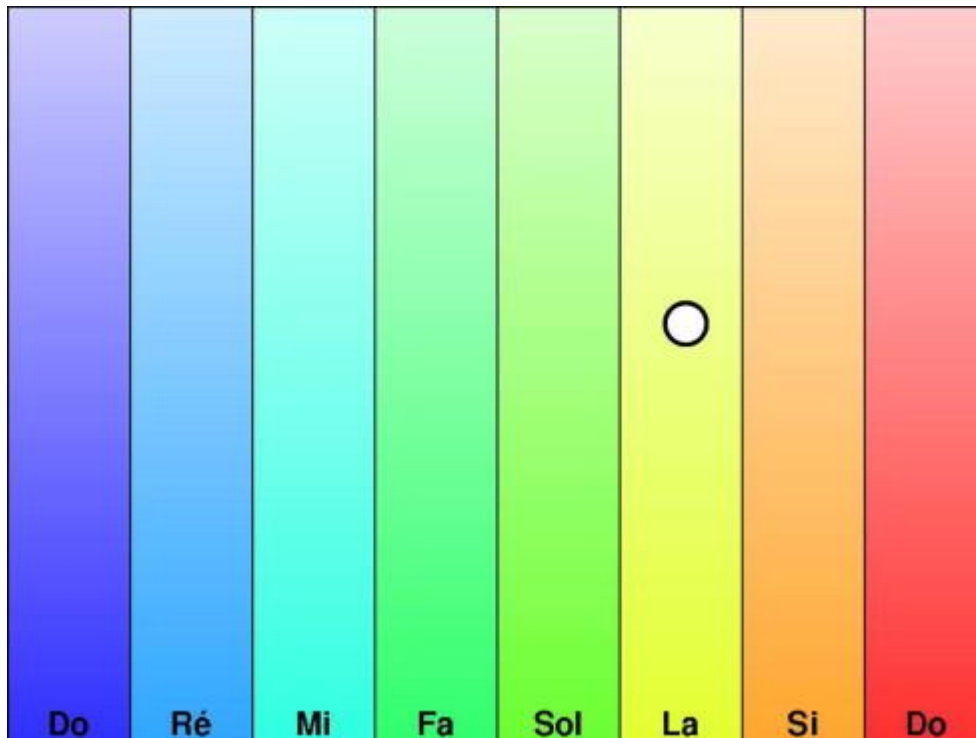


Figura 3 Captura de MINWii Project

- **Celeste:** plataformes 2D en el que la protagonista, qui pateix d'ansietat, fa un viatge fins la cim de la montanya Celeste.
 - Cercador: Google
 - Paraules Clau: mental health video game
 - Resultats: 411.000.000



Figura 4 Portada de Celeste

- **Hellblade: Senua's Sacrifice:** joc de puzzles i combats 3D on la protagonista pateix de psicosis.
 - Cercador: *Google*
 - Paraules Clau: *mental health video game*
 - Resultats: *411.000.000*



Figura 5 Portada de Hellblade

- **Multiple Sclerosis: Changes in Microarchitecture of White Matter Tracts after Training with a Video Game Balance Board:** article acadèmic que estudia l'entrenament amb videojocs per si poden induir canvis significants en connexions cerebrals i altres paquets associatius supratentorials.
 - Cercador *Google Scholar*
 - Paraules Clau: *multiple sclerosis video game*
 - Resultats: *15.400*

- **Re-Mission:** joc seriós per ensenyar a nens a entendre les malalties cancerígenes.
 - Cercador: *Serious Game Classification*
 - Paraules Clau: *disease*
 - Resultats: *26*

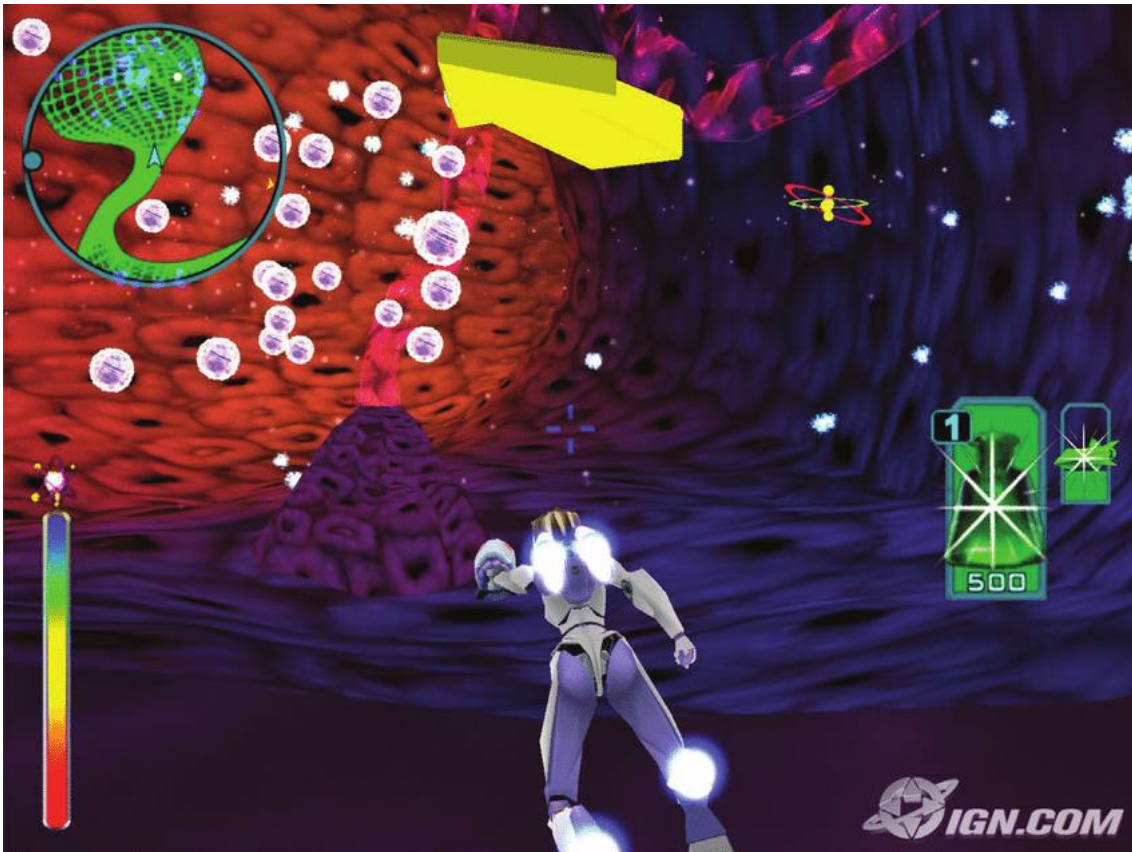


Figura 6 Captura de Re-Mission

2.5.3 MATRIU DE COMPETITIVITAT

A partir de les cerques realitzades i tenint en compte els objectius del videojoc proposat, s'han definit els següents requisits:

- Ha de ser un videojoc.
- Ha de ser lúdic, no seriós.
- Ha de ser un plataformes 2D.
- Ha de tractar sobre l'Esclerosis Múltiple.
- El públic objectiu ha de ser gent sense la patologia tractada.

La matriu de competitivitat resultant és la següent:

<i>Joc</i>	<i>Videojoc</i>	<i>Tipus</i>	<i>Plataformes 2D</i>	<i>Esclerosis Múltiple</i>	<i>Públic Objectiu</i>
<i>Medulla</i>	Sí	Seriós	Sí	No	No Patologia
<i>MINWii Project</i>	Sí	Teràpia	No	No	Patologia
<i>Celeste</i>	Sí	Lúdic	Sí	No	No Patologia
<i>Hellblade</i>	Sí	Lúdic	No	No	No Patologia
<i>Article EM</i>	No	-	-	Sí	-
<i>Re-Mission</i>	Sí	Seriós	No	No	No Patologia
<i>Joc Proposat</i>	Sí	Lúdic	Sí	Sí	No Patologia

Taula 3 Matriu de Competitivitat

2.5.4 CONCLUSIONS ESTUDI DE MERCAT

Com es pot veure amb l'estudi realitzat i la matriu de competitivitat, el joc proposat es diferencia força d'altres projectes amb temàtiques semblants. La majoria de jocs sobre patologies estan destinats a servir de teràpia per a pacients o bé són jocs seriosos per ensenyar la malaltia de forma interactiva. No és el cas que s'està buscant, ja que el joc proposat no busca ensenyar, sinó visibilitzar i que la gent que no pateixi la patologia es pugui posar en la pell de qui sí la tingui.

2.6 PÚBLIC OBJECTIU I PERFIL DE JUGADOR

Una vegada s'ha analitzat el mercat, s'ha d'aprofundir per veure com es dissenyarà el joc per tal d'atraure a un públic més gran. S'han de tenir clars alguns punts que poden ser difícils d'encarar per un dissenyador:

- **No ets el jugador:** no s'ha de pensar que el jugador serà com el dissenyador. S'ha d'imaginar que pot ser molt diferent i per tant s'ha d'analitzar diferents maneres perquè el jugador se senti còmode jugant el joc.
- **El jugador no és el teu oponent:** no es lluita contra el jugador. En dissenyar s'ha de pensar a entretenir el jugador, no fer-li perdre els nervis. Hi ha moltes maneres de fer això i és la funció del dissenyador pensar la millor manera.
- **Pensa en el jugador primer:** no es pot començar a dissenyar res del joc sense primer fer-se la pregunta de quina és la funció del jugador i què farà aquest quan tingui el joc entre les mans.

Per definir el perfil de jugador del joc proposat, s'analitzarà la tipologia de jugadors definida per Richard Burtle i que s'ha estudiat durant el curs. Segons Richard Burtle, es pot classificar l'estil de joc dels jugadors en 4 diferents tipus de perfil.

- **Achievers:** tenen com a objectiu resoldre el major número de reptes i dificultats. Obtenen el plaer resolent situacions complexes. Interessats a actuar en el món.

- **Explorers:** els hi agrada explorar a fons el videojoc, descobrir els secrets i aprendre coses desconegudes del videojoc. Interessats en interactuar amb el mon.
- **Socializers:** els hi agrada més interactuar amb altres persones que el propi rerefons del videojoc. Interessats a interactuar amb els jugadors.
- **Killers:** els hi agrada competir contra altres jugadors. Interessats a actuar sobre els jugadors.

Al jugador del projecte proposat li agrada completar els nivells i els reptes que conté. A més, com que s'haurà d'adaptar als símptomes de la malaltia, s'interessarà en resoldre les situacions complexes que el joc ofereixi. Per tant, el jugador que es busca és l'*Achiever*.

És també important fixar-se en l'edat d'aquest públic, ja que pot variar substancialment la manera com aquest reben els estímuls i reaccionen davant diverses situacions del videojoc. Per exemple, nens més petits poden acabar molt enfadats i acabar deixant el videojoc davant 3 partides seguides perdent o poden desinteressar-se si la idealització de les estratègies és massa complexa i se'ls hi escapa del seu abast. El jugador buscat és potencialment jove adult, ja que és el més interessat en temes mèdics i d'empatia, a part d'obtenir plaer amb els reptes proposats.

2.7 CONCLUSIÓ DE LA VIABILITAT DEL PROJECTE

Pel que fa al programari necessari, s'utilitza programari gratuït i versions d'estudiant dels que no ho són. Per tant el cost és 0.

Pel que fa a la remuneració dels treballadors, en el cas hipotètic d'una contractació real, tampoc seria del tot ajustada envers el que podem veure a la taula anterior, ja que aquest treball moltes de les hores han estat invertides a fer recerca, investigar i provar noves tecnologies de les quals no se'n tenia un coneixement previ. Amb això, la contractació de treballadors sèniors o professionals en el sector no hauria suposat un preu tan elevat en nombre d'hores de disseny, i segurament també menys en programació i art.

L'estudi de mercat realitzat mostra que el projecte seleccionat es viable en el mercat per l'originalitat en la proposta, i el públic objectiu és prou ampli com per ser competent en el mercat.

Com a conclusió, com que és un projecte d'estudiant i es disposa dels requisits tecnològics, el treball serà viable.

3 PLANIFICACIÓ

Aquesta etapa defineix l'estratègia seguida per arribar als objectius plantejats. Es descriurà breument el pla de treball, les tasques planificades, el temps estimat i els resultats esperats de cada tasca.

Se sol posar en concordança tots els membres de l'equip per treballar de forma eficient, es reparteix el treball segons les disciplines, es fixen els terminis d'entrega, es consensua cada quant es faran reunions per veure el progrés, etc. En aquest cas, com que es realitza de forma individual, les reunions són les que es faran amb el tutor del treball.

En un videojoc, com a qualsevol projecte es troben les 3 etapes fonamentals, l'etapa de disseny, l'etapa d'elaboració, i l'etapa de retoc i proves abans de la seva publicació. La diferència radica en què en un videojoc veiem que hi ha moltes parts en el desenvolupament que s'executen de forma iterativa, i per tant pot ser el cas que, després d'implementar quelcom, es torni a la fase de disseny un altre cop. Per tant es podria dir que en els videojocs hi ha fases de proves i errors i continus canvis mentre es desenvolupa el videojoc, tot i abans d'arribar a la vertadera fase de proves i errors.

Per aquest motiu també és molt important la figura del dissenyador, perquè és qui ha de definir molt exactament els objectius de l'equip així com totes les coses que són primordials en el videojoc i ja més tard i si es pot permetre complementar-lo amb coses secundàries. Un bon disseny al principi fa que aquesta part sigui més ordenada.

Així doncs, veiem que amb una bona planificació aconseguirem ordenar correctament les prioritats i organitzar-nos col·lectivament, i d'aquesta manera millorar en eficiència.

3.1 PLA DE TREBALL

En aquest apartat es plantejaran les coses que es faran, així com el temps que portarà dur-les a terme i quin serà el mètode a seguir per cada cosa. S'han de considerar tots els escenaris possibles, ja que poden sorgir molts problemes inesperats, així com acurtar temps en tasques que en un principi es podria pensar que portarien molt més temps (per exemple, la implementació d'una mecànica molt complicada però que en consultar internet en cerca de fonts d'informació trobar llibreries o eines amb la feina ja feta).

Els videojocs solen tenir un procés iteratiu per cada una de les tasques (Figura 7). Per exemple, es dissenya un personatge, es genera un prototip i s'avalua, i llavors potser tornem a la fase de disseny perquè hi ha alguna cosa que no queda com es pensava, i així fins que s'obté un resultat satisfactori.

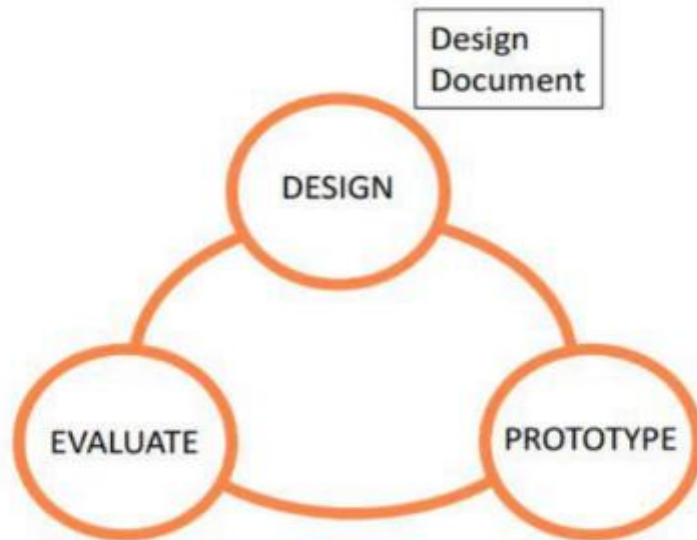


Figura 7 Procés Iteratiu

3.1.1 ABAST DEL PROJECTE

Per fer una bona planificació, primer s'ha de tenir molt clar l'abast del projecte. S'ha de tenir ben definides totes les coses que apareixeran en el videojoc, ja que així podem organitzar-nos per a fer un correcte anàlisi i fer una correcta extracció de requisits del projecte. Una vegada estiguin definits els punts més importants, diferenciar i classificar les diferents tasques a realitzar es torna molt més senzill.

A continuació s'exposa un recital de punts importants que ha de contenir el videojoc:

- Videojoc de Plataformes 2D.
- Té com a objectiu posar al jugador a la pell d'una persona amb EM.
- Tindrà uns pocs nivells on es mostraran diferents dels símptomes de la malaltia.
- Els controls bàsics seran els típics d'un plataformes 2D: desplaçar-se i saltar.
- La resta de mecàniques representaran accions que el jugador haurà de fer per adaptar-se als símptomes de la malaltia.
- El joc no tindrà enemics, només obstacles.
- Els símptomes que es representaran són els següents:
 - Visió doble
 - Fatiga
 - Espasticitat
 - Desequilibris

3.1.2 TASQUES PLANIFICADES

Ara que s'han presentat els punts més importants que conté el nostre joc, és més fàcil dividir-lo per parts i organitzar les tasques. A continuació, una taula de les tasques:

Número	Tasca	Temps estimat
01	Presentació proposta a neuròlegs especialitzats.	2 hores
02	Reunió amb doctor especialitzat i pacients	2 hores
03	Reunió amb infermer especialitzat	2 hores
04	Entrevista amb pacient d'Esclerosis Múltiple	2 hores
05	Preparació de l'entorn d'Unity. Crear una petita escena de proves, definir la càmera, etc.	6 hores
06	Desenvolupament en primera instància de personatge general. Objecte que es mogui com a màxim una certa distància, crear-li les físiques i <i>colliders</i> .	8 hores
07	Implementació de símptomes de la malaltia.	30 hores
08	Disseny de la protagonista.	6 hores
09	Animació de la protagonista.	12 hores
10	Disseny de nivells.	20 hores
11	2a reunió amb neuròleg especialitzat per obtenir <i>feedback</i> del procés.	2 hores
12	Implementació dels canvis aconsellats en la darrera reunió.	20 hores
13	Pulir <i>bugs</i> .	12 hores
##	TOTAL	102 hores

Taula 4 Taula de planificació

3.1.3 CRONOGRAMA

A partir de la temporalització associada a les tasques planificades s'ha fet un cronograma on es dóna una visió global de tot el projecte.

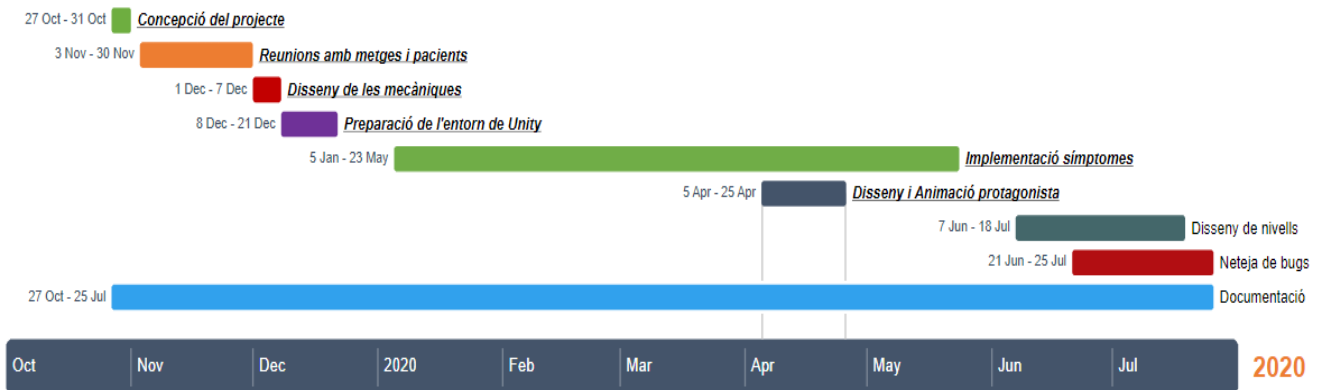


Figura 8 Cronograma de la planificació

4 MARC DE TREBALL I CONCEPTES PREVIS

Aquest punt servirà principalment per a preparar al lector i introduir els conceptes necessaris per a poder comprendre de forma correcta i fer un bon seguiment del document de disseny del videojoc.

D'aquesta manera, s'introduirà una mica el tipus de joc que es realitza. Això es farà exposant, primer, el màxim exponent del gènere i explicant-ne el *gameplay*. Després d'això s'exposarà quines diferències s'introdueixen en aquesta proposta de videojoc respecte al *gameplay* comentat, a l'igual que les similituds.

També s'explicarà quin motor s'ha decidit utilitzar per a desenvolupar el videojoc, conjuntament amb les raons per haver fet aquesta elecció.

En aquest apartat i com en realitat ja s'ha exposat en aquest document en anteriors moments, s'especifica que el treball és realitzat en la seva plenitud per una sola persona i la màxima participació de terceres persones en el disseny són les reunions realitzades amb neuròlegs i pacients que han influenciat molt en el disseny de les mecàniques.

4.1 REFERÈNCIES

La idea per a un nou videojoc pot sorgir de moltes maneres diferents. Pot ser que es tingui una idea completament nova que revolucioni un gènere o es pot imitar jocs consolidats i donar-los un cop de rosca. Pot haver la possibilitat que sorgeixi la idea d'un tipus de *gameplay* encara no definit i crear un subgènere propi en la indústria. En aquest treball, la idea sorgeix d'agafar una idea sortida d'una experiència personal i voler-la adaptar al gènere dels plataformes 2D.

A continuació s'explicarà el funcionament de dos jocs d'èxit que han servit d'inspiració per aquest projecte, i que a més tenen algunes similituds.

4.1.1 HELLBLADE: SENUA'S SACRIFICE

Hellblade: Senua's Sacrifice és un videojoc d'acció-aventura i terror psicològic, desenvolupat i publicat per Ninja Theory, llançat a tot el món de forma independent per a les plataformes PlayStation 4 i Microsoft Windows el 8 d'agost de 2017. Descrit per Ninja Theory com un "joc triple A independent", va ser desenvolupat per un equip d'aproximadament vint persones, amb escrits addicionals de la Dra. Elizabeth Ashman Rowe, professora universitària especialitzada en cultura escandinava.

Inspirat per la mitologia nòrdica i la cultura cèltica, Hellblade segueix a Senua, una guerrera celta que ha d'arribar a Helheim derrotant a entitats d'un altre món i enfrontant els seus reptes, per rescatar l'ànima del seu amant, mort pels vikings, de la deesa Hela. Paral·lelament, el joc actua com una metàfora de la lluita del personatge amb la psicosi, ja que Senua, qui pateix l'afecció però creu que és una maledicció, és perseguida per una entitat coneguda com "La Foscor", veus en el seu cap conegudes com "Fúries" i records del seu passat. Per representar correctament la psicosi, els

desenvolupadors van treballar en estreta col·laboració amb neurocientífics, especialistes en salut mental, organitzacions sense ànim de lucre i persones que pateixen aquesta afecció.



Figura 9 Captura del Hellblade

Fora de les seves escenes llargues i destacades, *Hellblade: Senua's Sacrifice* es divideix en dos tipus de jugabilitat: la primera li permet a Senua caminar lliurement i interactuar amb el seu entorn. Senua pot utilitzar una habilitat coneguda com "focus", en referència a la seva tendència a veure les coses de manera diferent a les altres persones per la seva condició, per desencadenar esdeveniments relacionats amb trencaclosques. Si usa el seu enfocament en els tòtems ocults durant el joc, desencadena un record a través de la veu en off del seu amic Druth, explicant-li les històries dels homes del nord. Diverses àrees presenten els seus propis mecanismes o mecàniques exclusives, com arribar a una zona segura en el temps abans que mori Senua, o utilitzar la capacitat d'enfocament per modificar l'estructura del seu entorn.

A més, *Hellblade* presenta combats en què Senua ha de vèncer a un o diversos éssers que bloquegen la seva progressió, inclosos els caps ocasionals. Durant aquestes baralles, Senua treu la seva espasa i es torna cap a un dels seus enemics, en què la càmera s'enfoca automàticament. Pot usar dos tipus d'atacs, ràpids i pesats, per expulsar un enemic i evitar que bloquegi els seus atacs, l'aturi o l'esquivi. També pot carregar a l'oponent en el qual s'està centrant actualment, per apropar-se o atacar directament.

Aquesta és una de les principals fonts d'inspiració del projecte, ja que l'objectiu del joc era visibilitzar la malaltia mental de la psicosis, i es va aconseguir gràcies a un assessorament mèdic especialitza.

4.1.2 CELESTE

Celeste és un videojoc pertanyent a el gènere de plataformes creat pels desenvolupadors de videojocs canadencs Matt Thorson i Noel Berry. El videojoc va ser creat originalment com un prototip en quatre dies durant una *gamejam*¹, i més tard va augmentar a un llançament complet. Celeste es va llançar al gener de 2018 per a les plataformes Microsoft Windows, Nintendo Switch, PlayStation 4, Xbox One, macOS i GNU/Linux.

Celeste és un videojoc de plataformes en el qual els jugadors controlen a una jove (l'edat no és esmentada) anomenada Madeline mentre puja una muntanya evitant diversos obstacles mortals. A més de saltar i grimpar parets durant un temps limitat, Madeline té la capacitat de realitzar una cursa enmig de l'aire en qualsevol direcció. Aquest moviment només es pot realitzar un cop i ha de reposar ja sigui aterrant a terra o colpejant certs objectes com reposar vidres. Al llarg de cada nivell, el jugador trobarà mecàniques addicionals, com molls que llancen a l'jugador o plomes que permeten un vol breu, i objectes mortals com pues que tornaran a Madeline a l'inici d'una secció si ella mor. Ocults al llarg de cada nivell hi ha maduixes opcionals, obtingudes a través de desafiadors plataformes o seccions de resolució de trencaclosques, que afecten lleugerament al joc.



Figura 10 Captura de Celeste

Com s'ha mencionat, Celeste segueix a una jove anomenada Madeline mentre intenta escalar la Muntanya Celeste, una muntanya fictícia a l'oest de Canadà que aparentment posseeix el poder de manifestar la confusió interna i el veritable jo en la realitat. Mentre

¹Event de desenvolupadors que té com a propòsit la creació d'un o més videojocs en un curt període de temps, normalment entre 24 i 48 hores.

puja la muntanya es troba amb altres personatges, inclòs Theo, un jove optimista i segur de si mateix de Seattle que va venir a la muntanya per fer fotografies; una dona gran que viu a la muntanya; El sr. Oshiro, el fantasma de l'conserge d'un hotel abandonat a la muntanya; i una altra Madeline hostil que és la part paranoica, pragmàtica i egoista d'ella, el cos ha estat creat per la muntanya. Madeline pateix de depressió i atacs de pànic, i mentre puja la muntanya, es veu obligada a enfrontar i superar els seus dimonis interns, a més de les proves que ofereix la muntanya.

Aquest joc també ha sigut una clara inspiració ja que va voler tractar un tema com la depressió i el va saber representar en un plataformes 2D.

4.2 MOTOR DE JOC: UNITY



El videojoc s'ha decidit implementar amb el motor de videojocs *Unity*, motor desenvolupat per *Unity Technologies*. A continuació s'explica perquè s'ha pres la decisió d'emprar *Unity* i també alguns conceptes essencials a conèixer sobre el motor.

En l'actualitat, la majoria de videojocs arreu del món són desenvolupats o bé per *Unity* o bé per *Unreal*, amb excepció de les grans empreses que es poden permetre tenir un departament sencer per a desenvolupar un motor propi. Segons la pròpia pàgina web d'*Unity*, esmenten que més de la meitat de videojocs del món són desenvolupats amb aquesta eina.

S'ha decidit emprar *Unity* perquè és una tecnologia que ja coneixia amb anterioritat perquè s'ha utilitzat durant el grau, i ofereix més comoditat per utilitzar gràcies a l'experiència obtinguda durant el curs. A continuació s'expliquen algunes característiques per les quals el motor ha esdevingut un dels millors, i en justifiquen l'elecció:

- Permet el desenvolupament multiplataforma per a gairebé totes les opcions que hi ha en el mercat, tant de consoles com dispositius mòbils. A més, exportar el mateix videojoc a diferents plataformes és molt fàcil de realitzar i ho fa a molta velocitat, deixant gairebé tota la feina a l'adaptació de controls i interfícies.

- Permet desenvolupar tota mena de videojocs, té diferents mòduls per treballar tant en 3D com en 2D. Integra exemples de molts tipus de gèneres i implementacions bàsiques per començar amb el desenvolupament.
- Té una versió gratuïta que és suficientment capaç per si mateixa. Es pot fer qualsevol videojoc sense cap restricció, però apareixen limitacions com que apareix el logo d'Unity en començar el joc, o els serveis que ofereixen estan limitats (per exemple, només permet tenir servidors dedicats de màxim 20 persones a la vegada).
- L'editor (el que es presenta en la Figura 11) és molt intuïtiu i senzill d'utilitzar. Permet treballar amb molts tipus de fitxers i per tant es pot complementar el motor amb molts programes externs com *Photoshop*, *Blender*, etc.

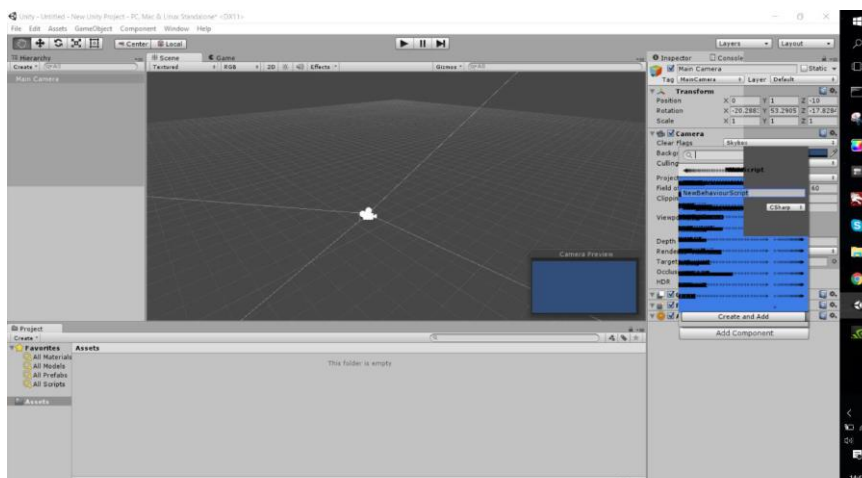


Figura 11 Captura de l'editor de Unity

- Té una gran documentació oficial, però possiblement i el més important, té una comunitat enorme de desenvolupadors. Hi ha molta informació a l'Internet i pràcticament es pot resoldre tots els dubtes sorgits cercant a la xarxa.
- El llenguatge de programació que empra és *c#*, el qual és força popular.

Per treballar amb *Unity* s'ha de conèixer uns conceptes fonamentals:

1. *Gameobjects*: cada objecte que és a l'escena. Això comporta tant personatges, com llums, l'escenari, etc. Els *gameobjects* per si mateixos no fan res, estan formats per **components**.
2. *Components*: aquí és on estan implementades les funcionalitats. Defineixen el comportament dels *gameObjects*.
3. *Variables*: els components tenen diferents paràmetres que es poden modificar, tant des de l'editor com a través d'*scripts*.

4.3 PEGI



Per a fer una correcta regulació dels videojocs, es va crear el sistema PEGI (Pan European Game Information). És un sistema d'autoregulació, és a dir, la pròpia desenvolupadora el defineix perquè el consumidor pugui saber quin tipus de contingut apareix i per a quina edat està recomanat. Després, però, passa per un procés de comprovació que entrega a l'editor la llicència que autoritza a utilitzar les etiquetes PEGI.

Per a tal comesa, es disposa de dos tipus d'ícones descriptores:

- Edat recomanada
- Contingut del joc

Així, abans de comprar un videojoc, el consumidor pot veure si el videojoc és apte per a ell mateix o per a qui va dirigit veient les etiquetes (Figura 12), i així pot utilitzar aquesta informació per a decidir si jugar-lo o no. En cap moment aquest indicador és prohibitiu, és a dir, encara que posi que el joc està recomanat perquè el juguin persones majors de 18 anys, una persona de 16 pot anar a una tenda i comprar-lo sense cap problema.



Figura 12 Etiquetes de PEGI

En el cas del projecte treballat, per l'estil del videojoc i pels elements que apareixen en el seu interior, s'hauria de senyalar que és **PEGI 12**.

5 ESCLEROSIS MÚLTIPLE

En aquest apartat es farà una introducció a la malaltia que ha estat la inspiració en aquest projecte. Es comentarà la patologia des d'un punt de vista mèdic, es parlarà dels símptomes i es mostraran algunes decisions de disseny que s'han dut a terme i com han estat influenciades per l'aprenentatge de l'EM, i de les reunions i entrevistes amb neuròlegs i pacients.

5.1 INTRODUCCIÓ

L'Esclerosis Múltiple és una malaltia autoimmune complexa, inflamatòria autoimmune, desmielitzant¹ i neurodegenerativa. Afecta a adults joves, sent la principal causa de discapacitat neurològica no traumàtica en aquest grup de pacients. És la 2^a causa de discapacitat en l'adult jove, per darrere dels accidents de tràfic. La seva naturalesa crònica comporta una elevada carga econòmica i social per l'individu, la seva família i la societat en conjunt.

Els símptomes poden començar de manera insidiosa i continuada; altres vegades, la seva evolució cursa amb brots². En qualsevol cas, els símptomes inicials poden ser molt diversos, abarçant des d'alteracions motores i sensibles, fins les alteracions neuroftalmològiques i el deteriorament cognitiu.

La malaltia afecta a unes 2.5 milions de persones en tot el món. Espanya ha passat a formar part dels països amb una prevalència alta d'EM. La prevalència estimada a Espanya és de 80-125 casos per cada 100.000 habitants per gènere i franja d'edat, cap a l'any 2009³. En l'última dècada s'ha observat un augment de prevalència i incidència d'aquesta patologia a Europa, en la conca mediterrània i presumiblement en tot el món.

És més freqüent en dones, amb un risc relatiu de 2.5:1 en relació als homes. Aquesta diferència ha augmentat considerablement en els últims anys a causa d'un dramàtic increment dels casos d'EM en el sexe femení. Les diferències solen igualar-se en edats més tardanes i les formes progressives solen ser més freqüents en homes.

5.2 SÍMPTOMES

L'EM afecta a cada persona de forma diferent, d'aquí a que es conegui també com "La Malaltia de les Mil Cares", i per això moltes vegades és difícil de diagnosticar. La manifestació característica de l'EM són els denominats brots, recaigudes o exacerberacions. En el 85% dels pacients, la malaltia s'inicia amb aquestes recaigudes.

El brot és un concepte clínic que representa l'activitat inflamatòria recurrent que passa en la substància blanca del Sistema Nerviós Central (SNC), que afecta a la mielina, els oligodendròcits i les axones. Es denomina fase remitent al període de temps lliure de símptomes neurològics entre brots. Els brots poden resoldre's amb o sense seqüeles.

¹Procés patològic en el qual es fa malbé la capa de mielina de les fibres nervioses.

²Símptomes d'aparició subaguda i autolimitada en el temps.

³Dades extretes de Manejo Del Paciente Con Esclerosis Múltiple En Atención Primària.

Un “pseudobrot” és un empitjorament o aparició de símptomes en presència de febre, augment de la temperatura o malaltia sistèmica.

Degut a que l'EM pot afectar a diferents àrees al llarg del SNC, els pacients pateixen d'una àmplia varietat de símptomes (Taula 5).

Símptomes Inicials EM	% dels pacients	Durant el transcurs d'EM
Visuals/Oculomotors	50-53	100
Motors	40-45	88
Sensitius	40-43	87
Cerebel·losos	20-25	82
Genitourinaris	10-13	63
Cognitius	5	39

Taula 5 Símtomes Inicials i Durant el Cours de l'EM

Els símptomes més freqüents són:

- **Visuals:** neuris òptica, diplòpia, oscilòpsia, oftalmoplèxia internuclear.
- **Motors:** pèrdua de força, dificultat per caminar, rigideza, espasmes musculars, problemes per l'articulació del llenguatge.
- **Sensitius:** sensació de formigueig, entumiment, esponjament, cremor, tibantor de cara, tronc o extremitats.
- **Cerebel·losos:** inestabilitat en la marxa, disartria cerebelosa, temblor, incoordinació motora dels membres.
- **Genitourinaris i sexuals:** urgència i incontinència urinària, retenció urinària, estrenyiment, incontinència fecal, dificultat per l'erecció, problemes per l'ejaculació, disminució de la lubricació vaginal, disminució de líbid.
- **Mental:** Alteracions de l'estat d'ànim, pèrdua de memòria, dificultat per concentrar-se, canvis emocionals, deteriorament cognitiu.
- **Fatiga:** sentir-se cansat, sense massa energia física o mental que no es correlacionen amb el grau d'activitat realitzada.
- **Paroxístmics:** neuràlgia del trigemin, episodis breus i repetits de dificultat per parlar, diplòpia, espasmes tòncics.

5.3 EXPERIÈNCIA AMB METGES I PACIENTS

Abans d'escriure la primera línia de codi o de dissenyar el primer nivell, es van fer reunions amb professionals especialitzats en la malaltia i pacients diagnosticats per obtenir un enriquiment i un aproximament mèdic i personal de la patologia i així poder estructurar i dissenyar el projecte amb més fidelitat.

La primera reunió va ser amb el Dr. Lluís Ramió i Torrentà, cap de Neurologia dels hospitals de Sta. Caterina i del Trueta. Amb ell es va fer la presentació de la idea del projecte, explicar com aquest videojoc està pensat per a visibilitzar la malaltia i, per tant, el públic objectiu és gent que no en sigui diagnosticada. El Dr. Lluís Ramió ens va fer una breu presentació de l'EM, diferenciant-la d'altres malalties degeneratives com L'Esclerosis Lateral Amiotròfica. També ens va proporcionar bibliografia per obtenir més formació de la patologia, a més de contactes d'altres professionals que ens poguessin ajudar.

La següent reunió va ser amb el Dr. René Robles, neuròleg de l'hospital Sta. Caterina. En aquesta sessió, l'autor de l'obra va ser present en un seguit de reunions amb pacients que venien a fer revisió i control sobre la malaltia a l'hospital. Amb l'autorització dels pacients, es va tenir els primers contactes amb diagnosticats on explicaven quines dificultats tenien: problemes amb la mobilitat dels dits, vèrtig, incontinència, visió doble, etc. Els pacients van ser majoritàriament dones, algunes de les quals havien estat diagnosticades d'EM entre els 20 i 30 anys i portaven amb la malaltia uns 10-15 anys.

Una altra reunió va ser duta a terme amb l'infermer Miquel Merchan de l'hospital Sta. Caterina. Aquest professional havia fet recentment una presentació on parlava de com visibilitzar símptomes de l'EM de forma interactiva com efectes òptics o jocs. L'infermer, llavors, va fer un resum d'aquesta presentació, va explicar els diferents tipus de símptomes (mencionats en el subapartat anterior) i com ell els transmetia de forma interactiva. En aquesta sessió es va discutir sobre com utilitzar els videojocs com a eina per a visibilitzar i donar immersió sobre malalties i de quines formes es podria fer.

Finalment es va fer una entrevista amb la Begonya Matesanz, pacient diagnosticada d'EM. En aquesta sessió, La Begonya va explicar la seva experiència amb la malaltia, com va ser diagnosticada i les diferents experiències amb símptomes i brots. Va parlar de la seva descordinació motora, la dura fatiga en el començament dels dies i l'aparició d'espasmes amb el cansament, entre d'altres. També va comentar la seva manera de lluitar contra la patologia, com l'ús del Yoga.

5.4 ESCLEROSIS MÚLTIPLE EN EL VIDEOJOC

Després de les reunions i entrevistes, es va haver de transformar l'après de la malaltia i els símptomes en una estructura de videojoc amb unes mecàniques concretes. A continuació es farà el llistat dels símptomes seleccionats i perquè es van triar.

FATIGA

És el símptome més comú en tots els pacients i el que més pateixen en el seu dia a dia, i per tant es va decidir que havia de tenir un lloc en el joc. Es va transformar el símptome en una barra d'energia, inspirat en les barres d'estamina de molts videojocs com *Dark Souls* o *Zelda: Breath of the Wild*, en els quals certes accions com atacar o esprintar gasten energia, i l'esgotament d'aquestes barres obliguen al jugador a descansar.



Figura 13 Captura del Dark Souls 3. La barra verda de l'esquina superior esquerra és l'estamina, la qual es gasta quan el jugador ataca, bloqueja atacs enemics o esprinta.

VISIÓ DOBLE

Un tema molt important que es tenia present durant la formació de la malaltia era quines accions feien els pacients per combatre l'aparició d'alguns dels seus símptomes. En una de les primeres reunions amb el Dr. Lluís Ramió, se'ns va comentar com amb l'aparició de visió doble, alguns pacients es tapaven un ull per reduir l'efecte. Això va inspirar una mecànica que, quan sorgís aquest símptome en la partida, el jugador es veíés obligat a mantenir un botó per a veure correctament, creant certa incomoditat deguda a l'ergonomia del mando i que transmetés la molèstia d'haver de tapar-se un ull per realitzar altres accions (en aquest cas, saltar obstacles).

ESPASTICITAT

Molts dels pacients amb els quals es va tenir contacte patien d'aquest símptome, el qual es manifesta com una rigidesa en alguns músculs. Es va pensar que, en un gènere de videojocs enfocat a saltar plataformes i superar obstacles, un símptome que reduís la mobilitat del jugador afegiria una dificultat molt present de la malaltia.

DESEQUILIBRIS

Un altre símptome molt freqüent entre els pacients d'EM és el dels desequilibris a l'hora de caminar. Els afectats d'aquest símptome solen adoptar formes de caminar més adients (passos més petits) i han de fer un esforç constant per no desequilibrar-se. D'aquí va sorgir una inspiració estreta directament d'un videojoc en concret on existeix una mecànica per mantenir l'equilibri. En aquest joc, *Death Stranding*, el protagonista ha de transportar paquets a través d'un mapa, i en cas de que pes dels paquets desequilibri al personatge, el jugador haurà de mantenir els dos disparadors del mando per no caure al terra. Pel projecte treballat, s'ha agafat aquesta mateixa mecànica transformada en una barra de desequilibri, la qual s'anirà emplenant a mesura que el jugador es mogui i concluirà amb la caiguda de la protagonista si no s'evita mantenint pres es botons d'equilibri.



Figura 14 Captura del *Death Stranding*

A part de l'elecció de símptomes, les reunions van tenir pes a l'hora de certes eleccions narratives. Per exemple, es va triar que la protagonista fos una dona d'uns 30 anys perquè és la franja de sexe i edat que més casos dona d'Esclerosis Múltiple.

6 DISSENY DEL VIDEOJOC

A continuació es presenta el capítol que, en realitat, cobreix les parts més importants del document de disseny del videojoc. S'han omès les parts que han estat comentades en un altra secció, com per exemple el públic objectiu i perfil del jugador, o com l'estudi de mercat.

6.1 ESTIL

S'ha decidit començar per l'estil, per donar a entendre ràpidament el concepte general del joc i que el lector pugui començar a imaginar-se'l en el cap. En aquest subapartat s'expliquen els diferents aspectes que en conjunt definirien l'estil del videojoc. Trobarem comentats el gènere, l'estil visual, la perspectiva i posició de la càmera.

L'estil moltes vegades és igual o més important que el propi gameplay, perquè és la imatge del joc, però és també qui determina el funcionament d'aquest en molts casos, i defineix moltes funcionalitats. Per posar un exemple, si l'estil està definit com per a jugadors petits, podem simplificar molt les físiques dels objectes.

6.2 GÈNERE

El joc proposat pertany al gènere de plataformes 2D. Els plataformes són un gènere de videojocs que es caracteritzen per haver de caminar, córrer, saltar o escalar sobre una sèrie de plataformes i penya-segats, amb o sense enemics, mentre es recullen objectes per poder completar el joc. Aquest tipus de videojocs solen usar vistes de desplaçament horitzontal cap a l'esquerra o cap a la dreta. És un gènere molt popular de videojocs, sorgit a començament de la dècada de 1980 i que segueix mantenint bastant popularitat en la actualitat, sent un dels més influenciadors, encara a dia d'avui, el *Super Mario Bros* de *Nintendo*.

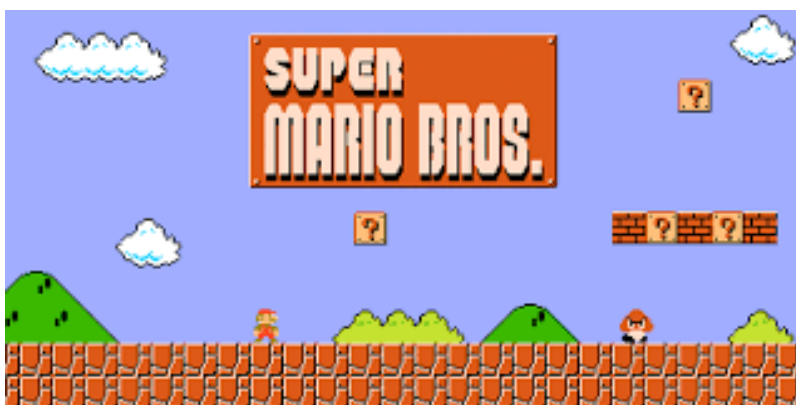


Figura 15 Super Mario Bros, per la Nintendo NES

Hi ha plataformes 2D amb sistemes de combat; d'altres, com *Super Mario Bros*, utilitzen les pròpies mecàniques del gènere, com el salt, per acabar amb els enemics; i n'hi ha que els reptes només venen de les plataformes. El joc proposat pertany a aquest últim grup, sense enemics i centrat en les plataformes i en els reptes sorgits pels símptomes de la patologia.

6.2.1 CÀMERA

La càmera que s'utilitza en el videojoc és la possiblement més utilitzada al llarg de la història de la indústria: és la perspectiva 2D i frontal que es veu en la Figura 16. Cal destacar que "juga" amb avantatge perquè altres que empenen el 3D han estat limitades a la potència tecnològica del moment. Aquesta perspectiva ha estat sobretot molt utilitzada per videojocs de plataformes.

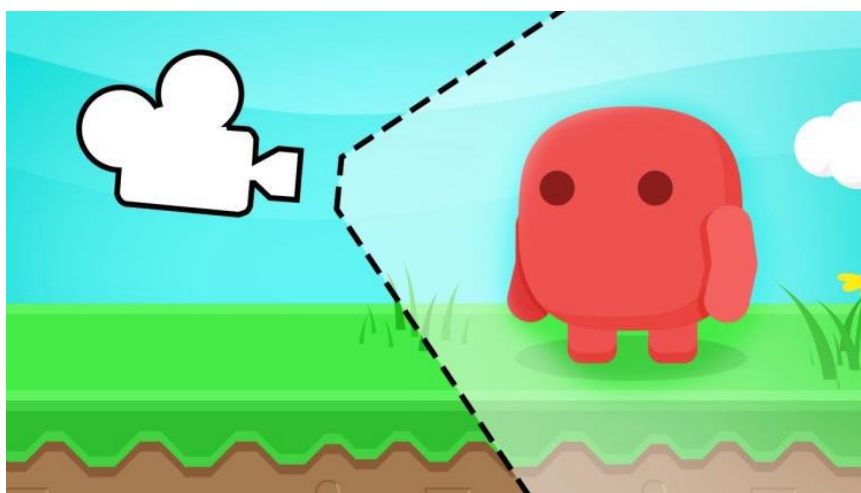


Figura 16 Representació de la càmera 2D frontal

6.2.2 GRÀFICS

Amb aquest apartat es vol fer referència al tipus d'estil artístic del qual disposarà el videojoc. Igual que ha passat amb diferents subapartats del capítol actual, i en realitat com sol ocórrer sempre sigui quin sigui el videojoc, un gènere no determina l'estil visual del joc, ni els gràfics que aquest tindrà.

En el projecte treballat es busca un estil gràfic *pixel art*, gènere artístic digital nascut de les limitacions tecnològiques de l'època d'on van sorgir els primers videojocs, i conegut per les seves imatges editades a nivell de pixel.



Figura 17 Exemple de pixel art fet a mà

El disseny de la protagonista i la seva animació han estat realitzats per l'autor de l'obra utilitzant el programa online *Piskel*¹. La resta d'art ha estat recollit d'internet amb llicència gratuïta per a la utilització, o extrets dels *assets* de lliure utilització propis de Unity.

6.3 GAMEPLAY

El *gameplay* és la font de l'entreteniment en tots els videojocs. És el més important i la primera cosa que s'ha de tenir en compte a l'hora de dissenyar un videojoc. Les mecàniques són l'element que defineixen el *gameplay* del videojoc. El *gameplay* consisteix en els reptes que el jugador ha d'encarar i les accions que el jugador pot realitzar per tal de superar-los.

6.3.1 DEFINICIÓ DE REPTES

El repte principal és una cosa tan general i simple com guanyar la partida, i d'aquest en surten diferents subconjunts de reptes que, amb les accions del jugador, es poden superar per assolir l'objectiu principal. Després s'explicarà d'una forma més entenedora gràcies a la jerarquia de reptes, on es veuran els diferents tipus de reptes des del més simplificat de tots, els quals s'anomenaran com reptes de nivell atòmic (no divisibles), fins als de les capes superiors anomenats no-atòmics (formats per diferents reptes).

A l'hora de fer els reptes, el dissenyador s'ha de posar en la pell del jugador. El prototip treballat requereix un balanç en el disseny molt acurat, ja que el joc proposa donar al jugador dificultats relacionades amb els símptomes de l'Esclerosis Múltiple per crear reptes afegits a l'hora d'afrontar les plataformes, i per tant és fàcil crear una experiència frustrant i que acabi aburrint al jugador, fent que deixi de jugar.

¹<https://www.piskelapp.com/>

6.3.2 JERARQUIA DE REPTES

Com s'ha dit anteriorment, el repte és molt clar: com a la majoria de plataformes 2D, és arribar al final del nivell. Però a part de superar els obstacles trobats, també s'afegeixen els reptes sorgits pels símptomes de la malaltia. En el següent diagrama (Diagrama 1) es pot veure el funcionament.

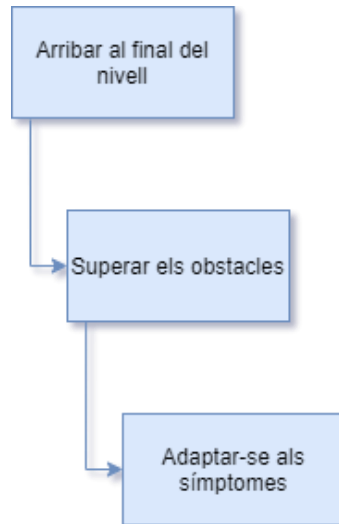


Diagrama 1 Jerarquia de reptes

Cada símptoma, però, afegeix un repte en específic. A continuació es mostra quin repte suposa cada símptoma:

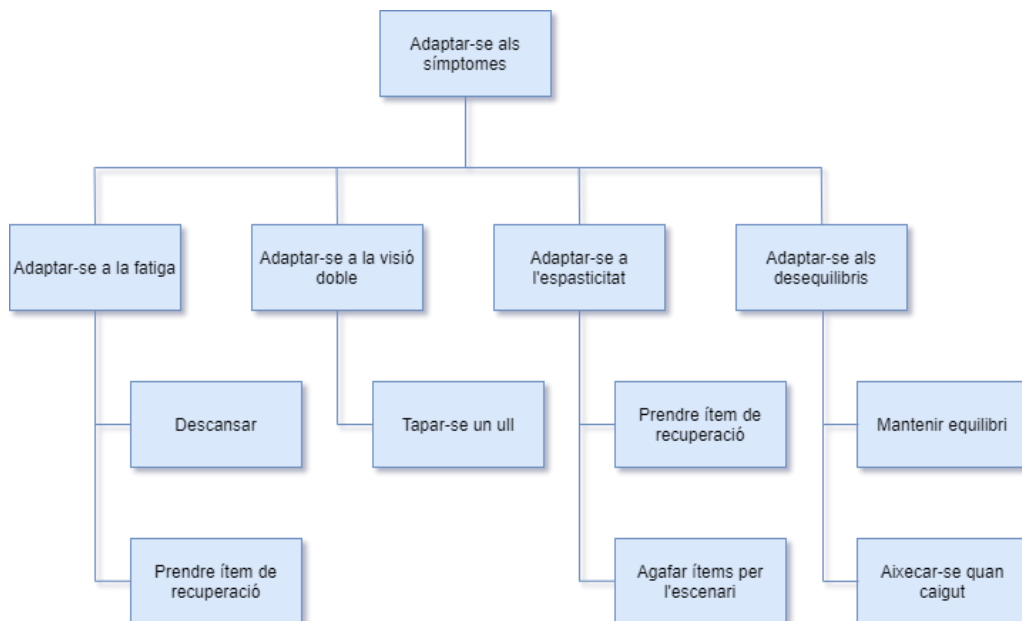


Diagrama 2 Jerarquia de reptes dels símptomes

6.3.3 DEFINICIÓ D'ACCIONS

Són, com bé diu el nom, les accions que un jugador pot fer en el món del joc, com poden ser saltar, atacar, parlar, etc. Per definir-les, es basaran en els reptes que hi ha en el joc, principalment en els de nivell inferior o reptes atòmics, i es pensarà en l'acció que permetrà superar cadascun d'ells.

També hi ha accions que normalment no van associades a cap repte, que poden ser accions per interactuar amb l'història del joc o el món del joc, accions per controlar l'estat del joc com per posar pausa, etc.

En el videojoc treballat hi ha les següents accions per superar els reptes:

- Moure's: desplaçar-se cap a l'esquerra o la dreta.
- Saltar: impulsar-se cap a munt.
- Aixecar-se: tornar-se a posar dret després de caure al terra.
- Tapar-se un ull: mantenir un ull tapat per millorar la visió doble.
- Utilitzar ítem: consumir un recurs de recuperació.
- Agafar ítem: agafar un recurs de recuperació.
- Mantenir equilibri: evitar els desequilibris.

6.4 MECÀNIQUES

Les mecàniques no són molt diferents de les normes o instruccions que es poden trobar, per exemple, en un joc de taula. El que cal saber sobre les mecàniques és que són el que generen el *gameplay*:

- Defineixen els tipus de reptes que el videojoc pot oferir.
- Defineixen les accions que el jugador pot fer per superar aquests reptes.
- Determina l'efecte de les accions del jugador envers el món del joc.
- Determina les condicions per aconseguir els objectius del joc i les conseqüències de tenir èxit o fallar en aconseguir-los.

Bàsicament, són les dades i algorismes que en un joc defineixen les lleis del joc i les operacions internes. En aquest joc, les mecàniques són principalment els símptomes implementats de l'Esclerosis Múltiple. És la part de disseny on s'ha posat el major esforç i on gira la major part del procés creatiu i el que s'ha volgut fer més fidel segons l'assessorament mèdic de les entrevistes i reunions amb metges i pacients.

6.4.1 RECURSOS

A continuació s'explicarà els recursos que s'han dissenyat pel videojoc. Els recursos són objectes del joc o materials que es poden crear, utilitzar, destruir, etc. Són el que en programació s'entenen com a classes.

Personatge

Vides
Recuperadors Energia
Velocitat
Força de Salt
Doble Salt (booleà)
Espasticitat (booleà)
Visió Doble (booleà)
Fatiga (booleà)
Desequilibri (booleà)

Taula 6 Recurs 'Personatge'

Fulla (Recuperació Espasticitat)

Recuperació

Taula 7 Recurs 'Fulla'

Nivell

Escenari
Càmera
Canvas
Grid

Taula 8 Recurs 'Nivell'

El disseny està compost pels següents recursos:

- Personatge jugable, qui controla el número de vides, les velocitats de moviment, força de salt, si tindrà l'habilitat de fer doble salt i quins símptomes tindrà. Es tracta d'un recurs tangible ja que es pot identificar en una partida i té propietats físiques en el món.
- Fulla, la qual quan s'utilitza, recupera del símptome de l'espasticitat.

- Nivell, el qual té l'escenari, la càmera, el canvas que controla les interfícies, i el grid de tilemaps que formarà el nivell. Es tracta d'un recurs intangible ja que representa les dades corresponents a l'estat del joc.

6.4.2 ENTITIES

Els *entities* són les instàncies d'un recurs o l'estat d'algún element del món del joc.

Recurs: Personatge	Entitat: PlayerStart
Vides	3
Recuperadors Energia	3
Velocitat	20
Força de Salt	35
Doble Salt	Cert
Espasticitat	*
Visió Doble	*
Fatiga	*
Desequilibri	*

**Els símptomes estaran a Cert o Fals depenent del nivell.*

Taula 9 Entitat 'PlayerStart'

Recurs: Fulla	Entitat: Leaf
Recuperació	100

Taula 10 Entitat 'Leaf'

6.4.3 DESCRIPCIÓ DE MECÀNIQUES

Fins ara s'han descrit les diferents mecàniques que governen el joc de forma general, explicant els elements que es trobaran en una partida i les seves característiques. En aquest apartat es descriuran les diferents mecàniques del joc detalladament.

6.4.3.1 OBJECTIUS

L'objectiu del joc és finalitzar tots els nivells; cada un d'aquests nivells es completa arribant al final superant tots els obstacles i plataformes. Si el jugador cau al buit, perdrà una vida i reapareixerà a l'últim punt de control. A continuació, totes les condicions per les quals el jugador haurà de començar el nivell des del principi:

- Perdre totes les vides.
- Quedar-se sense energia (síntoma fatiga) i no tenir ítems de recuperació.
- Estar afectat per l'espasticitat i no tenir ítems de recuperació.

6.4.3.2 CONTROLS BÀSICS

Tot i que l'aplicació del prototip del joc treballat sigui per PC, el joc s'ha dissenyat per jugar-lo amb un mando ja que la seva ergonomia i el posicionament de botons han de servir per transmetre millor els símptomes que es representen. Per tapar-se l'ull, el jugador ha de mantenir pulsat l'RB, i per mantenir l'equilibri haurà de tenir pulsats l'LT i RT alhora. Això crea una incomoditat al jugador que se suma als reptes del nivell, volguent representar les dificultats afegides que té una persona amb EM.

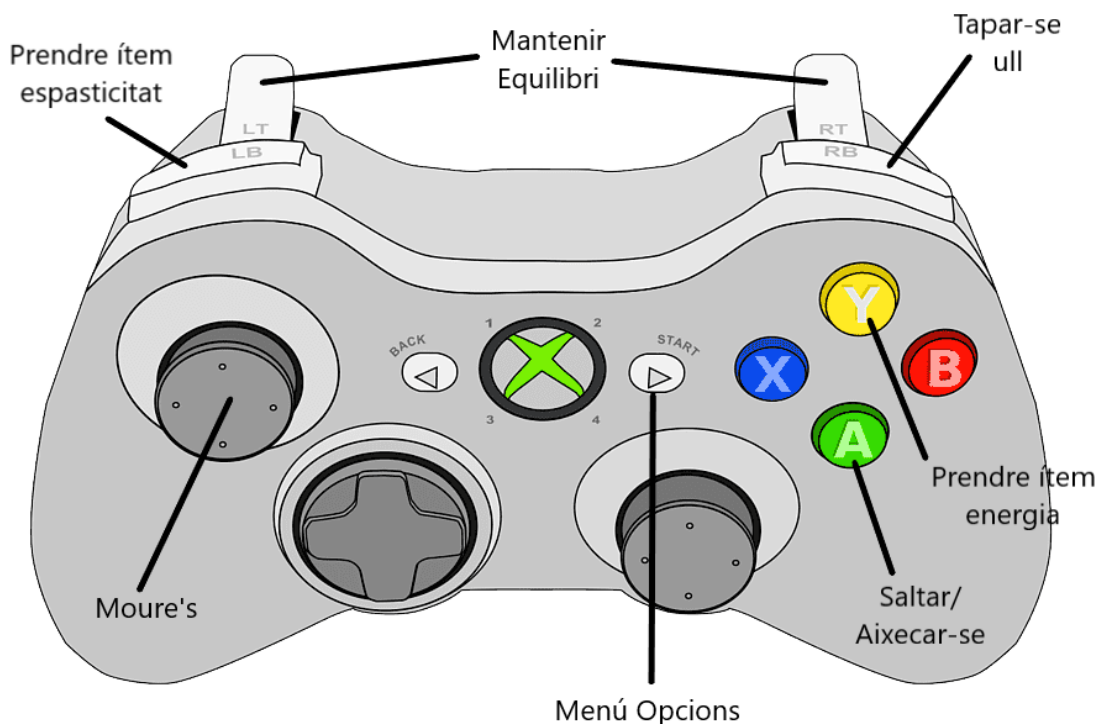


Figura 18 Controls

Tot i que el joc està pensat per jugar amb mando per les raons esmentades anteriorment, també s'han configurat els controls per a jugar-lo amb el teclat d'un PC. Els controls són els següents:

- Moure's: A i D (esquerra i dreta respectivament).
- Saltar/Aixecar-se: Espai.
- Mantenir Equilibri: Fletxes esquerra i dreta.
- Tapar-se ull: Fletxa avall
- Prendre ítem energia: E
- Prendre ítem espasticitat: Q
- Menú Opcions: ESC

6.5 FLOWCHART

El *flowchart* és un tipus de diagrama que representa un procés o un flux de treball. En el Diagrama 3 es pot veure el flowchart del joc, ensenyant les diferents pantalles que hi ha hi com s'accedeixen a elles.

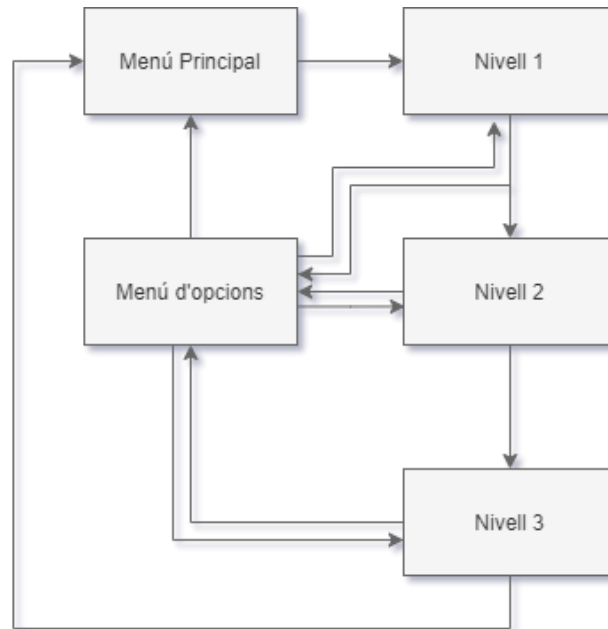


Diagrama 3 Flowchart del joc

Des del menú principal s'inicia el joc, començant pel primer nivell. Cada nivell superat porta automàticament al següent. Una vegada superat tots els nivells, es retorna al menú principal. Des de qualsevol nivell es pot accedir al menú d'opcions, des del qual es pot sortir de la partida i tornar al menú principal, o retornar al nivell que s'estava jugant.

6.6 LEVEL DESIGN

Normalment, quan es parla de disseny de nivells es pensa en una successió de nivells en les quals s'ha dividit un joc o una història. En aquest cas, pel disseny de nivells bàsicament ens referim als diferents escenaris que trobarem dins el videojoc.

El disseny d'escenari és molt important, perquè pot decidir el tipus de reptes que es poden trobar. En el cas d'aquest projecte, no s'ha volgut buscar una major complexitat en els nivells ja que es vol buscar el repte a través dels símptomes, no exclusivament per les plataformes, pel qual un disseny més complex podria frustrar al jugador.

6.6.1 ECONOMIA INTERNA

L'economia és el sistema pel qual recursos i entitats són produïts, consumits, i intercanviats en quantitats quantificables. A continuació s'explica com els diferents elements del joc són controlats.

FATIGA

La fatiga es representa amb una barra que en començar el joc està al 100% d'energia. Cada vegada que el jugador salta i cada segon que estigui caminant, aquesta barra anirà baixant. Quan baixi fins al 50%, seran reduïts tant la velocitat de desplaçament com la força de salt d'una forma no excessiva però sí significativa, de forma que el jugador sàpiga que moure's té conseqüències. Quan arribi al 10%, el desplaçament i el salt es reduiran al mínim, d'una forma que l'experiència sigui casi injugable. Al 0%, el personatge no es podrà moure.

Si el jugador descansa (no es mou ni salta), llavors recuperarà energia lentament. Depenent de la fase de cansament on es trobi, la recuperació serà més ràpida o més lenta:

- Entre el 50% i el 99% d'energia, la recuperació serà més ràpida.
- Entre el 10% i el 49%, serà una mica més lenta.
- Entre l'1% i el 9%, serà significativament més lenta.
- Al 0%, l'única forma de recuperar-se és prenent un ítem de recuperació.

El jugador començarà amb un número limitat d'aquest ítem i haurà de gestionar bé l'energia per no gastar-la i quedar-se sense recuperacions.

ESPASTICITAT

L'espasticitat és un símptoma de l'EM que no es pot predir quan afectarà al malalt. De tal forma, en el joc apareixerà de forma aleatòria cada cert rang de temps cada cop. Es representa amb un número entre el 0 i l'1, sent 0 no patir l'efecte i 1 no poder-se moure. En aquest prototip, però, només s'intercalarà la variable entre 0 i 0.5, punt en el qual el jugador es mourà de forma reduïda. L'única forma de recuperar l'estat normal és utilitzant un recurs especial i limitat de cada partida. Com que aquest símptoma és aleatori i imprevisible, en l'escenari del nivell hi haurà distribuïts ítems de recuperació que el jugador podrà agafar per no quedar-se sense.

VISIÓ DOBLE

La visió doble estarà definida per la "distància" que hi hagi entre una vista amb l'altra. Aquesta distància estarà gestionada en els scripts i augmentarà amb el temps. Si s'aplica la mecànica de tapar-se un ull, llavors la distància retornarà a 0, corresponent amb un dels dos ulls.

DESEQUILIBRIS

El desequilibri es representa amb una barra que, en començar el joc, està al 0%. Cada vegada que el jugador salta i cada segon que estigui caminant, aquesta barra anirà augmentant. Si arriba al 100%, el personatge caurà i haurà de fer una acció per aixecar-se. Mentre es mantinguin pulsats els dos botons (LT i RT) per mantenir l'equilibri, la barra baixarà i quedarà estable.

VIDES

Un nivell es comença amb un número predeterminat de màxim de vides. Cada vegada que es caigui al buit, baixarà en un el número de vides. Si arriba al 0, el jugador començarà el nivell des del principi, un altre cop amb el màxim de vides.

RECUPERADORS ENERGIA

Un nivell es comença amb un número predeterminat de recuperadors d'energia. Quan el jugador es quedi a 0% d'energia no es podrà moure, i la solució serà prendre un ítem de recuperació.

RECUPERADORS ESPASTICITAT

Un nivell es comença amb un número predeterminat de recuperadors d'espasticitat, però es podran agafar més durant la partida, els quals estaran repartits pel nivell. Quan el jugador pateixi d'espasticitat, el qual arribarà de forma aleatòria cada cert temps, haurà de recuperar-se prenent un ítem.

6.7 CANVIS REALITZATS DE DISSENY

Després d'haver començat a implementar i programar un primer prototip del videojoc treballat, es va fer una reunió de seguiment amb el Dr. Lluís Ramió. En aquesta reunió, se li van ensenyar els avanços fets fins el moment, ja que el seu *feedback* en aquest estat de desenvolupament era important per saber si s'estava recreant la malaltia amb fidelitat. Alguns d'aquests canvis han estat realitzats pel prototip final i s'explicaran a continuació.

- En un principi, la fatiga funcionava de forma binària: el personatge es movia de forma normal fins que arribés al 0 d'energia, on llavors li costaria desplaçar-se.
 - Se'ns va recomenar representar la fatiga en fases, de manera que la dificultat en moure's vagi escalant a mesura que l'energia va baixant.
- Es va aconsellar que quan s'activés la visió doble, també hi hagués una borrositat en la imatge, ja que una visió doble nítida era poc realista.

- Per donar més immersió sobre el que pateix una persona amb EM, es va proposar simular un tremolor quan la protagonista estigués a prop de tenir un desequilibri.

6.8 HISTÒRIA

La història, en aquest estat prototip de desenvolupament, serveix com excusa estètica i narrativa per a que el jugador vagi del punt A al punt B del nivell.

La protagonista, la qual no té nom, és una dona d'uns 30 anys del Japó feudal que pateix d'Esclerosis Múltiple. Cada cert temps, fa una peregrinació a un temple situat a dalt de tot d'una montanya. Però cada vegada que la fa, es troba amb noves dificultats degudes als nous brots de la patologia. Aquestes peregrinacions són les que en el videojoc es representen com nivells.

6.8.1 MÓN DEL JOC

DIMENSIÓ FÍSICA

El prototip contindrà nivells que representaran un tros acotat de la peregrinació, però no tota la peregrinació sencera. Estarà dissenyat com un camí 2D on desplaçar-se, tan horitzontal com verticalment. Tots els nivells estaran ambientats en la montanya que porta al temple.

DIMENSIÓ TEMPORAL

La protagonista fa aquesta peregrinació cada cert temps; temps suficient per a que en la malaltia apareguin brots nous amb símptomes nous, els quals poden ser de diversos anys. Cada nivell estarà ambientat en una peregrinació nova i, per tant, en una etapa de la patologia nova.

DIMENSIÓ AMBIENTAL

L'estètica del joc és japonesa feudal, tal i com s'ha descrit anteriorment. El disseny de la protagonista està pensat per a que recordi l'època asiàtica antiga en la que està ambientat el joc.

DIMENSIÓ EMOCIONAL

L'emoció més present en el transcurs de la història és l'autosuperació. En començar el projecte, abans de fer la 1a reunió amb els neuròlegs especialitzats, es va optar per un concepte de joc més pessimista degut a malentesos amb altres malalties degeneratives.

Després d'aquestes reunions, es va decidir fer un joc optimista respecte la patologia, ja que la societat d'avui en dia està prou preparada per a que una persona amb EM pugui tenir una vida normal, exceptuant per les dificultats afegides, i per tant és important donar un missatge esperançador.

6.9 INTERFÍCIE GRÀFICA

En aquest capítol s'explica el conjunt d'escenes que té el videojoc i també es detallen els elements més importants que veiem en el camp de joc. Primer, començarem amb les escenes i menús explicant-los breument i tot seguit veurem els elements de la interfície gràfica.

6.9.1 MENÚ PRINCIPAL



Figura 19 Menú Principal

És el menú que es troba quan s'obre l'aplicació. En aquest menú, el jugador podrà començar la partida o sortir del joc i tancar l'executable.

6.9.2 NIVELL

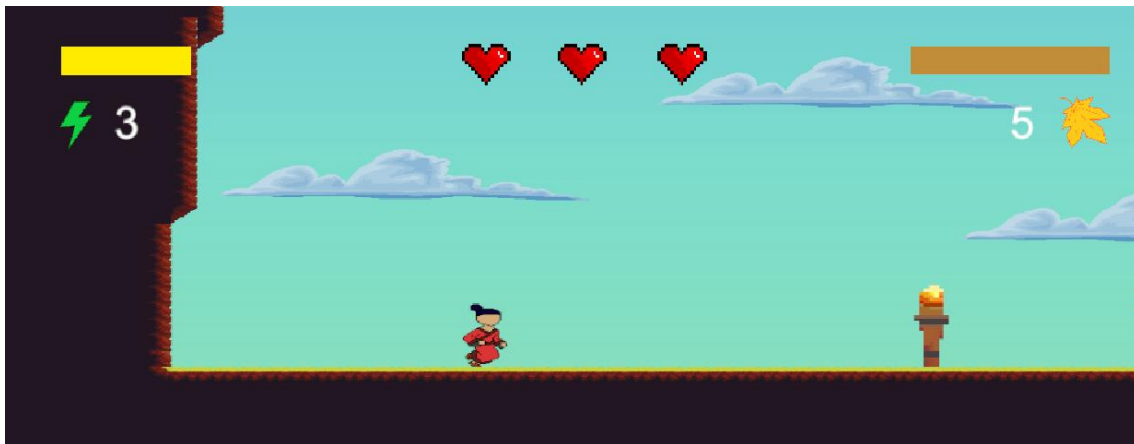


Figura 20 Nivell i interfície

La imatge mostra el que el jugador veurà en una partida. A continuació es descriurà cada element de la interfície.

ENERGIA



Figura 21 Barra d'energia i contador de recuperacions.

En aquesta imatge es pot apreciar tant la barra d'energia com el contador de recuperacions. La barra d'energia es troba a la part superior de la imatge, representada com una barra verda. Aquest és l'estat de la barra quan el jugador té el 100% d'energia. A mesura que es gasta, es va encongint. Quan arriba al 50%, la barra es tornarà groga; quan baixi del 10%, serà vermella.

El raig és l'ícona de les recuperacions, i el número de la dreta és la quantitat que se'n té.

VIDES



Figura 22 Vides del jugador

Les vides del jugador es mostraran amb tres cors a la meitat superior de la pantalla. Cada vegada que es perdi una vida, el nombre de cors es reduirà en un.

DESEQUILIBRI



Figura 23 Barra de desequilibri

El desequilibri es mostra com una barra marronosa que s'anirà emplenant a mesura que el jugador es mogui i salti.

FULLES



Figura 24 Contador de Fulles

Les fulles representen els recuperadors d'espasticitat. Estaran a la part superior dreta de la pantalla. El número de l'esquerra representa la quantitat de fulles disponibles.

PERSONATGE

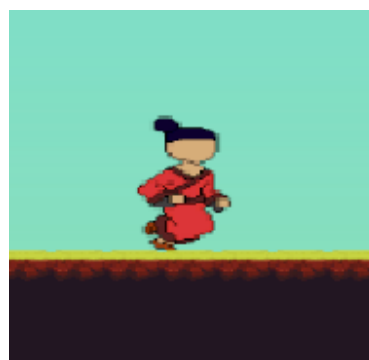


Figura 25 Protagonista

La protagonista es situa aproximadament al centre de la pantalla.

6.9.3 MENÚ OPCIONS

És el menú que apareix quan es pulsa el botó *Start* en un nivell. Des d'aquest es pot anar al Menú Principal, o retornar al nivell que s'estava jugant en el mateix moment on s'havia deixat.



Figura 26 Menú d'Opcions

6.10 ART DEL JOC

En aquest projecte, s'ha treballat en el disseny i l'animació de la protagonista per l'art del joc, a més d'un parell d'sprites d'interfície. A continuació es mostrarà tot l'art del joc, tant el que s'ha produït com el que s'ha extret per internet.

6.10.1 ESCENARIS

Per crear l'escenari s'ha agafat un *tilemap*¹ d'internet², ja que faciliten molt la feina i són molt útils per crear nivells en 2D ràpidament. El tilemap utilitzat és el següent:

¹Sistema de Unity que emmagatzema i gestiona els mosaics per crear nivells 2D

²<https://learn.unity.com/tutorial/live-session-2d-platformer-character-controller#>

Extret de: unity Learn

Autor: Matt Shell

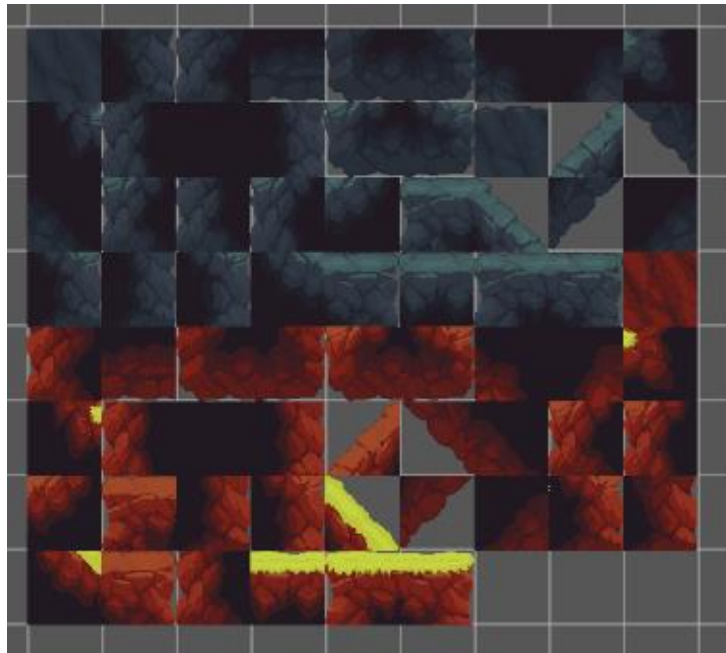


Figura 27 Tilemap utilitzat

Del mateix paquet gratuït de Unity d'on es va treure el *tilemap*, també es va treure el *background* i alguns elements decoratius de l'escenari com núvols o torxes.

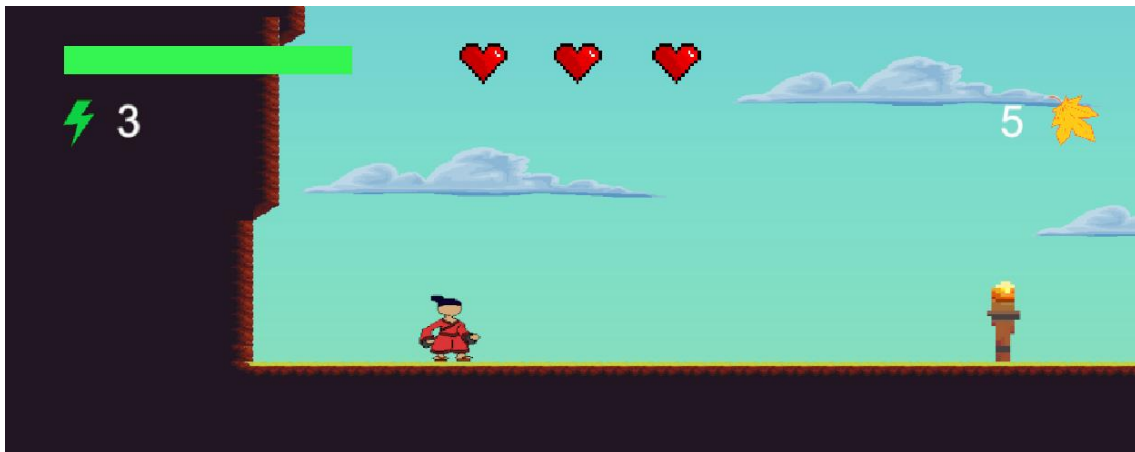


Figura 28 Captura de l'escenari de joc

Per separat:



Figura 29 Background

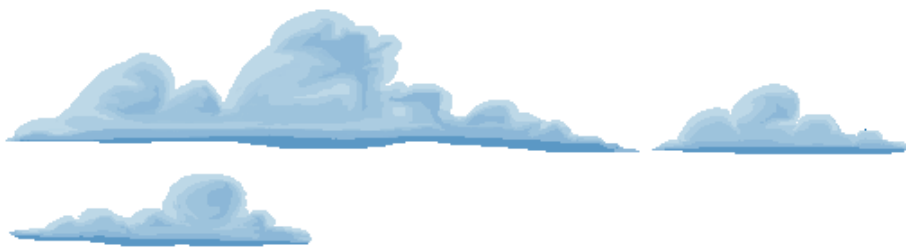


Figura 30 Núvol



Figura 31 Torxa

6.10.2 PERSONATGES

El joc té un únic personatge, la qual serà dissenyada primer a mà i, després de seleccionar el disseny final, es passarà a l'estil *pixel art* gràcies al programa Piskel.

ESBOSSOS

Figura 32 Esbossos Iniciais

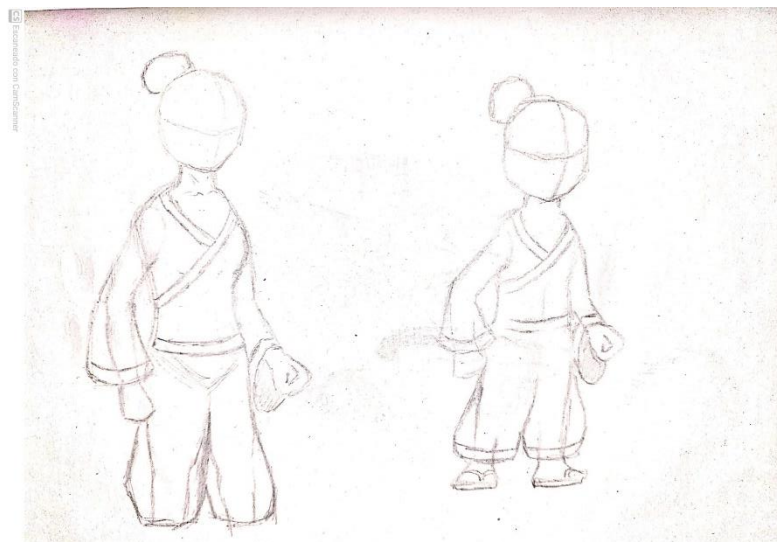


Figura 33 Esbossos finais

PIXEL ART



Figura 34 Disseny final en pixel art

ANIMACIÓ CÒRRER

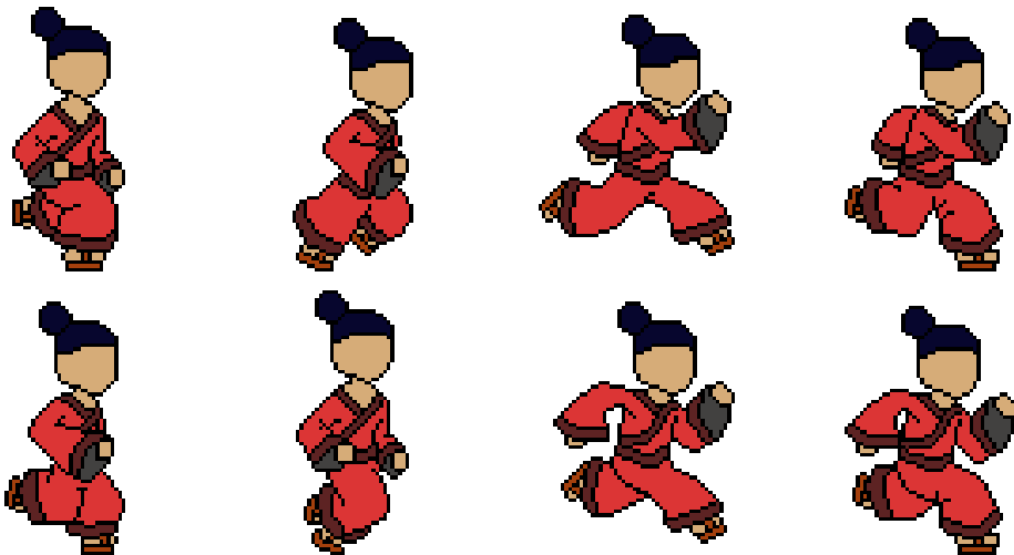


Figura 35 Sprites de l'animació de córrer

En aquest prototip, no hi ha animació *idle*¹ ni de salt. Quan està quieta, es queda en l'Sprite de la Figura 34. Quan salta, es fa servir un dels Sprites de l'animació de la Figura 35.

¹Animació típica dels videojocs 2D que apareix quan el personatge està quiet.

6.10.3 ALTRES SPRITES

El símbol del raig d'energia i la fulla, que representen els recuperadors d'energia i d'espasticitat respectivament, han estat produïts per l'autor de l'obra amb Paint 3D (veure Figures 36 i 37). El cor en pixel art s'ha extret d'internet (veure Figura 38).



*Figura 36 Raig
Energia*



Figura 37 Fulla

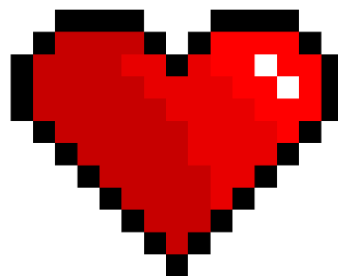


Figura 38 Cor pixel art

Extret de:

<http://pixelartmaker.com/art/9c1cbb90d7a7474>

6.11 EFECTES DE SO

El videojoc, com a mínim el prototip, no consta d'efectes sonors, ja que s'ha prioritzat altra feina abans. En tot cas, en no tenir les eines ni personal que tingui la capacitat per generar-lo, es disposaria a utilitzar sons i música de llicència oberta.

7 IMPLEMENTACIÓ I PROVES

En aquest apartat es comentarà tot el que s'ha pogut generar com a demostració jugable del projecte. És un prototip per veure reflectida la idea en general, pel qual hi ha moltes característiques que no s'han arribat a desenvolupar i el que es veu no tindria res a veure amb la versió final. En aquest prototip es presentaran tres nivells que representen trossos d'un nivell més gran hipotètic que no s'ha arribat a desenvolupar. També es pot veure una idea de com són les interfícies i el moviment entre menús. En aquest apartat no s'ensenya codi, aquest anirà adjunt en l'annex, és un apartat per entendre el disseny del *gameplay*.

A continuació es comenta diferents aspectes de la implementació i de com està estructurat el joc. Com s'ha mencionat anteriorment, el videojoc està format amb Unity i per tant l'estructura i el tipus de programació té un determinat estil propi d'Unity.

Per començar, explicarem l'estructura de carpetes del projecte.

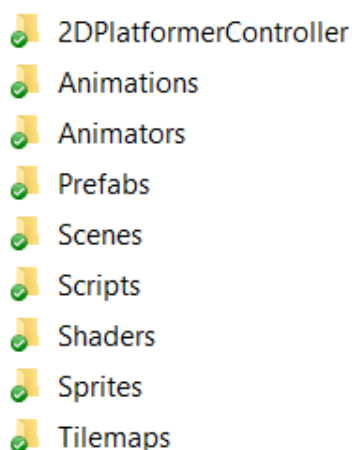


Figura 39 Estructura de Carpetes del Projecte de Unity

S'ha dividit el tipus d'objecte tractat en diferents carpetes per a poder organitzar millor tots els *assets* i poder accedir més fàcilment a ells. S'ha creat carpetes per a les coses més essencials, com els *scripts* o els *sprites*, per exemple. Dintre d'aquestes també hi ha subdirectoris per a tenir-ho tot molt ben organitzat. Per exemple, dins de *sprites* tenim diferents carpetes per si es tracta d'un personatge, un *Render Texture* o un *Tile*.

La 1^a carpeta mostrada a la Figura 39, *2DPlatformerController*, és el paquet de Unity baixat d'internet d'on s'han extret els *tilemaps* i alguns altres *sprites*.

7.1 ENTORN

A continuació s'explicarà com està format un nivell i quins elements el componen. A la Figura 40 es pot veure la jerarquia principal d'objectes de l'escena i seguidament es comentaran un a un.



Figura 40 Jerarquia d'objectes en l'escena d'un nivell

En una partida d'un nivell existeixen dos estats: en joc i en pausa. Quan s'està jugant, l'objecte Game, que conté tots els elements del nivell, està actiu mentre que CanvasOptions està inactiu. Quan s'està en pausa és al revés. A més, es troba el LevelController, que és el que controla en quin estat està la partida (joc o pausa) i a quina escena s'anirà quan s'acabi el nivell. Finalment tenim l'EventSystem, necessari per a interaccionar amb botons en els canvas.

A continuació, es mostren algunes de les funcions del LevelController.cs.

```
public void StartGame()
{
    SceneManager.LoadScene("Scenes/Level1");
}

public void QuitGame()
{
    Application.Quit();
}

public void Continue()
{
    Game.SetActive(true);
    Canvas.SetActive(false);
}

public void MainMenu()
{
    SceneManager.LoadScene("Scenes/MainMenu");
}

private void OnTriggerEnter2D(Collider2D collision)
{
    SceneManager.LoadScene(nextLevel);
}
```

CANVASOPTIONS

En l'estat de pausa, el jugador es trobarà amb el següent:



Figura 41 Captura de l'estat de pausa

L'estructura del CanvasOptions és la següent:

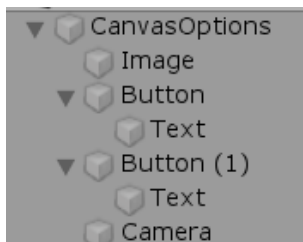


Figura 42 Jerarquia del CanvasOptions

L'objecte pare és un Canvas, i els seus fills són la imatge de fons, dos botons (Continue, Quit) i una càmera.

GAME

En l'estat de joc, el jugador es trobarà amb el següent:

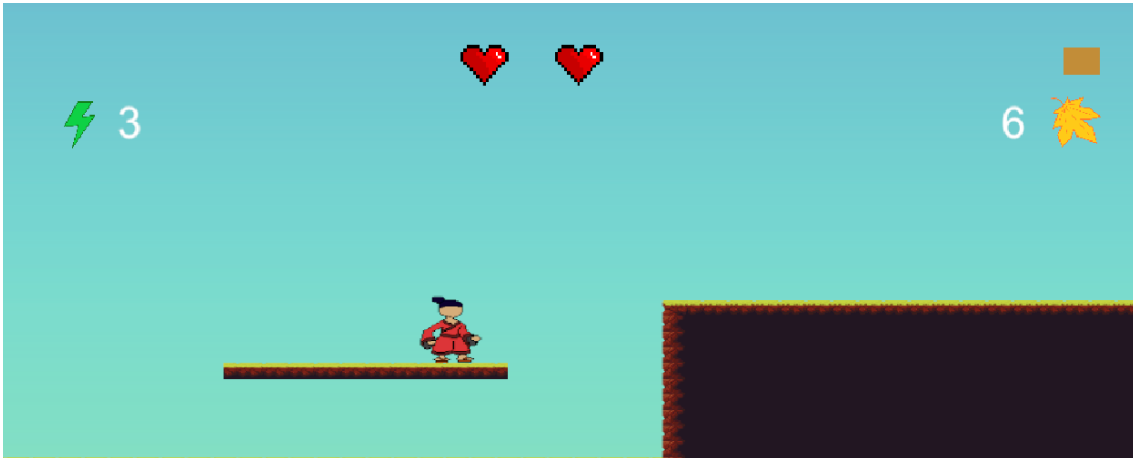


Figura 43 Captura de l'estat de joc

L'estructura de Game és la següent:



Figura 44 Jerarquia de Game

Com es pot veure en la Figura 44, l'objecte Game és el que conté la major part de l'escena. A continuació es comentarà cada objecte fill de Game i la seva funció.

- PlayerStart: és la protagonista i, per tant, l'objecte controlable de la partida.
- Main Camera: càmera principal de la partida.
- CameraRenders: càmeres auxiliars, que tenen assignades un *Render Texture* cada una, les quals serviran per representar el símptoma de la visió doble (s'explicarà detalladament més endavant).
- Canvas: conté tota l'interfície de la partida.

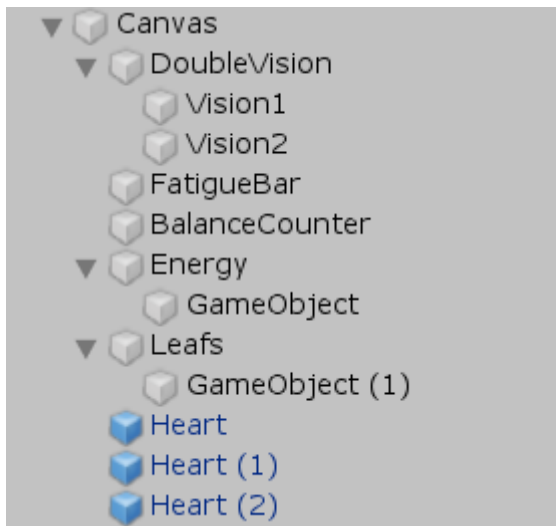


Figura 45 Jerarquia del Canvas

- DoubleVision: imatges que serviran per representar la visió doble.
 - FatigueBar: barra de fatiga.
 - BalanceCounter: Barra d'equilibri.
 - Energy: contador de recuperadors d'energia.
 - Leafs: contador de recuperadors d'espasticitat.
 - Hearts: vides del jugador.
- Grid: conté el *tilemap* del nivell.
 - Respawn0: Lloc on apareixerà el jugador quan comenci el nivell.
 - Deadzones: zones on el jugador perdrà una vida, com per exemple el buit d'un penyasegat.
 - Torchs: antorxes que serviran de punts de control.
 - Decoration: elements decoratius del nivell.

7.2 PERSONATGE

La protagonista del joc es controla amb l'objecte de l'escena `PlayerStart`. Conté els següents elements:

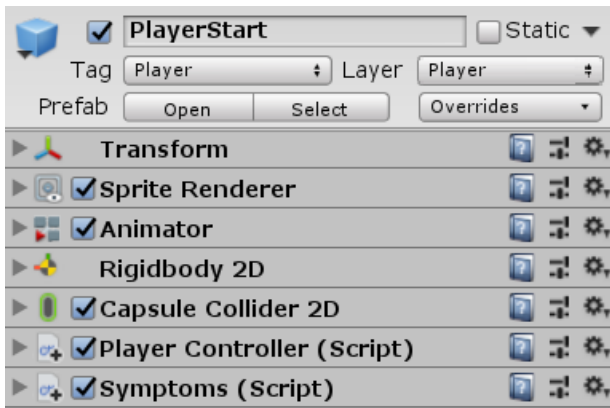


Figura 46 Elements de PlayerStart

- Transform: posició, rotació i escala de l'objecte.
- Sprite Renderer: sprite de la protagonista (Figura 34).
- Animator: controlador de les animacions.
- Rigidbody 2D: controla les físiques del personatge.
- Capsule Collider 2D: controla les col·lisions del personatge.
- Player Controller: Script dels controls del personatge.
- Symptoms: Script que controla els símptomes del personatge.

7.2.1 ANIMACIONS

La protagonista només té una animació, la qual és la corresponent als sprites de la Figura 35. Els estats que conté l'animador (còrrer, saltar, idle) es representen de la següent forma:

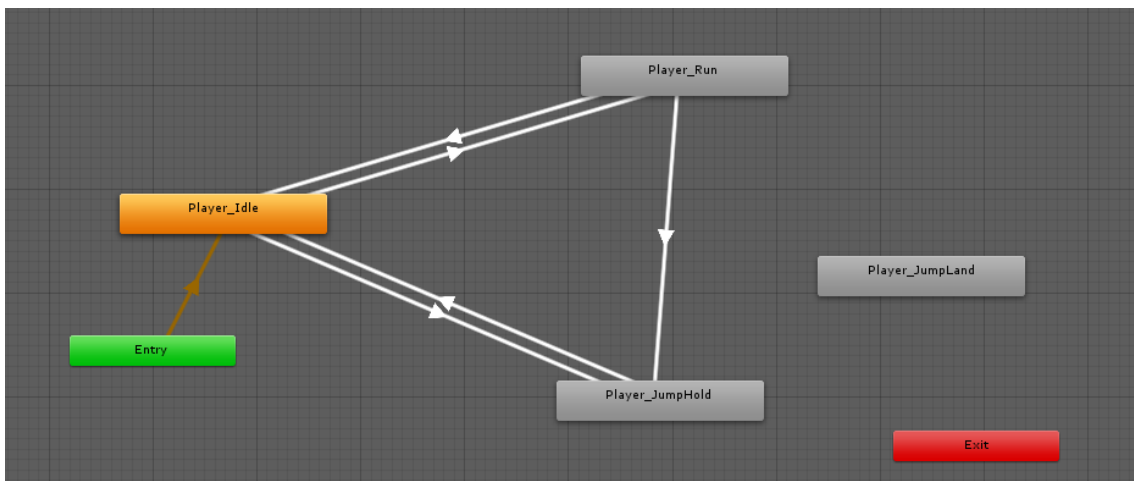


Figura 47 Animator de PlayerStart

Comença en Player_Idle. Si es desplaça va a Player_Run i si es para retorna a Player_Idle. Sempre que salti, tant si està en Player_Idle com en Player_Run, anirà a Player_JumpHold.

Del salt no podrà anar directament a Player_Run, haurà de passar primer abans per Player_Idle.

7.2.2 PLAYER CONTROLLER

L'script que gestiona els controls del personatge és PlayerController, com el desplaçament, el salt o la pausa. La vista des de l'inspector d'aquest script és la següent:

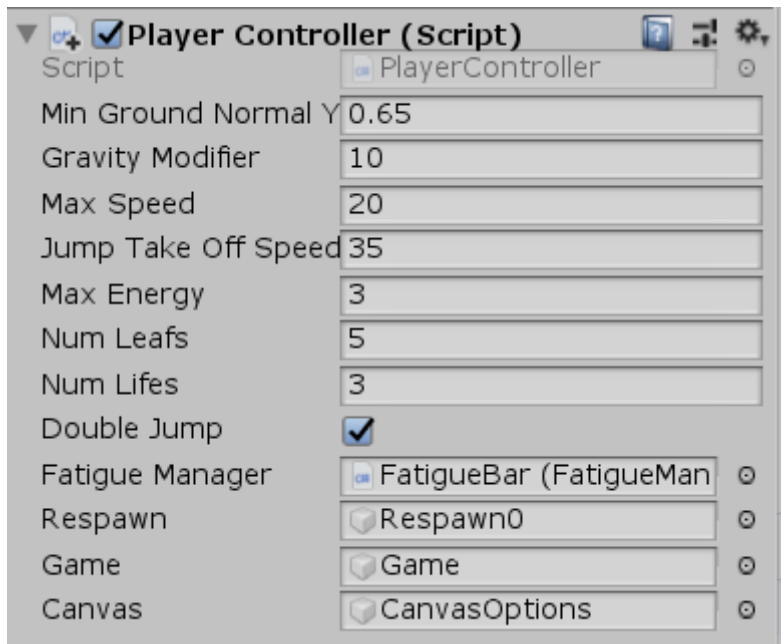


Figura 48 Script de PlayerController vist des de l'Inspector

- Min Ground Normal Y: controla l'angle de la superfície a partir del qual es considerarà que el personatge lliscarà cap avall.
- Gravity Modifier: gravetat del personatge.
- Max Speed: velocitat màxima que es pot anar.
- Jump Take Off Speed: força de salt.
- Max Energy: número màxim de recuperadors d'energia.
- Num Leafs: número de recuperadors d'espasticitat.
- Double Jump: booleà que controla si es pot fer doble salt o no.

A continuació, el codi de la funció principal del PlayerController.cs on es veu com gestiona els controls.

```

protected override void ComputeVelocity()
{
    if (Input.GetButtonDown("StartButton"))
    {
        Canvas.SetActive(true);
        Game.SetActive(false);
    }

    if (!symptoms.imbalanced)
    {
        Vector2 move = Vector2.zero;

        move.x = Input.GetAxis("Horizontal");
        if (move.x != 0)
        {
            moving = true;
            symptoms.imbalanceRange += 0.05f;
            if (grounded)
                fatigueManager.AddFatigue(0.025f);
        }
        else
            moving = false;

        if (Input.GetButtonDown("Jump") && grounded)
        {
            velocity.y = jumpTakeOffSpeed;
            fatigueManager.AddFatigue(2.5f);
            moving = true;
        }
        else if (Input.GetButtonDown("Jump") && !grounded && jump)
        {
            velocity.y = jumpTakeOffSpeed;
            jump = false;
            fatigueManager.AddFatigue(5f);
            moving = true;
        }
        else if (Input.GetButtonUp("Jump"))
        {
            if (velocity.y > 0)
            {
                velocity.y *= 0.5f;
            }
            moving = false;
        }

        if (!moving)
        {
            if (fatigueManager.FatigueBar <= 50 && fatigueManager.FatigueBar
> 10)
                fatigueManager.AddFatigue(-0.02f);
            else if (fatigueManager.FatigueBar <= 10 &&
fatigueManager.FatigueBar > 0)
                fatigueManager.AddFatigue(-0.01f);
            else if (fatigueManager.FatigueBar < 100 &&
fatigueManager.FatigueBar > 50)
                fatigueManager.AddFatigue(-0.1f);
        }

        if (grounded && doubleJump)
            jump = true;
    }
}

```

```

bool flipSprite = (spriteRenderer.flipX ? (move.x > 0.1f) : (move.x <
-0.1f));
if (flipSprite)
{
    spriteRenderer.flipX = !spriteRenderer.flipX;
}

/**BALANCE CONTROL**/
if ((Input.GetAxis("BalanceLeft") == 1 &&
Input.GetAxis("BalanceRight") == 1)
    || (Input.GetKey(KeyCode.LeftArrow) &&
Input.GetKey(KeyCode.RightArrow)))
{
    symptoms.balancing = true;
    //symptoms.imbalanceRange = 10;
}
else if (Input.GetAxis("BalanceLeft") < 1 ||
Input.GetAxis("BalanceRight") < 1)
{
    symptoms.balancing = false;
}

/**RECOVER ENERGY**/
if (Input.GetButtonDown("RecoverEnergy"))
{
    if(maxEnergy > 0)
    {
        fatigueManager.AddFatigue(-100);
        maxEnergy--;
    }
}

/**RECOVER SPASTICITY**/
if (Input.GetButtonDown("RecoverSpasticity"))
{
    if(numLeafs > 0)
    {
        symptoms.RecoverSpasticity();
        numLeafs--;
    }
}

animator.SetBool("grounded", grounded);
animator.SetFloat("velocityX", Mathf.Abs(velocity.x) / maxSpeed);
animator.speed = 1 - symptoms.spasticity;

targetVelocity = move * maxSpeed;
}
else
{
    animator.SetBool("grounded", grounded);
    animator.SetFloat("velocityX", 0);
    if (Input.GetButtonDown("Jump"))
    {
        symptoms.balancing = true;
        symptoms.imbalanceRange = 10;
    }
}
}
}

```


7.2.3 SYMPTOMS

En aquest script es gestionaran quins símptomes estaran activats en un nivell. També és on estan programats com aquests es representen.

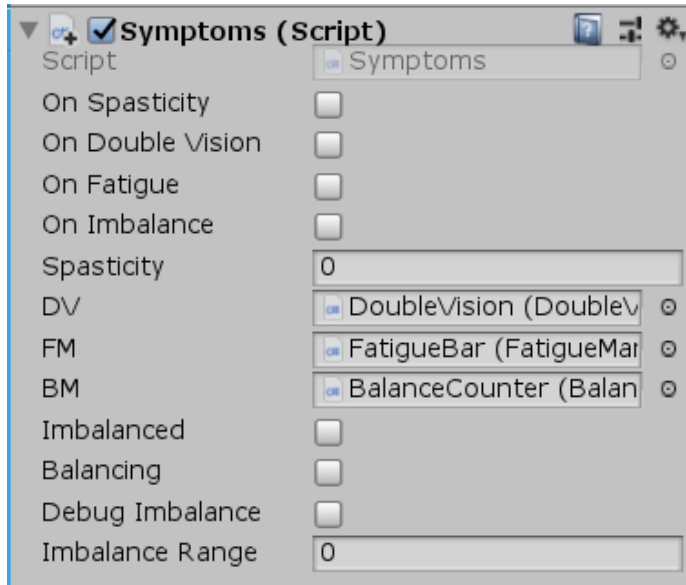


Figura 49 Script de Symptoms vist des de l'Inspector

Les primeres quatre variables que es veuen a la Figura 49 són els booleans que controlen quins símptomes estaran actius en el nivell. També s'hi troben tres scripts que formen part de les interfícies necessàries per alguns d'aquests símptomes. Altres variables són per controlar certs valors que s'utilitzaran públicament en altres scripts, entre d'altres de *debug*.

En aquest projecte ha sigut important la implementació dels brots, ja que són etapes molt comunes en el desenvolupament de la malaltia, i és on solen aparèixer símptomes nous. És per això que des de l'inspector de Unity es decidiran quins símptomes estaran actius i quins no a cada nivell, per simular un brot en concret en cada cas.

7.3 IMPLEMENTACIÓ CÀMERA

La càmera d'aquest projecte és molt semblant a la resta de càmeres típiques del gènere, la qual se situa sempre al mateix costat del personatge i el segueix vertical i horitzontalment.

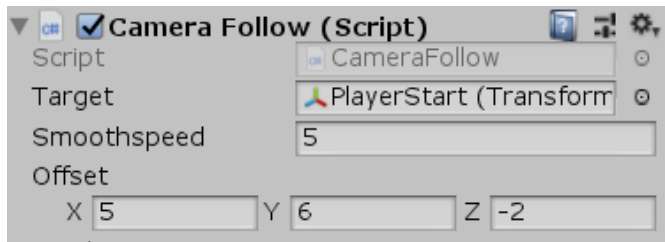


Figura 50 Script de la Càmera des de l'Inspector

L'script agafa un *Target*, en aquest cas el *PlayerStart*, el qual seguirà durant tot el nivell. L'*Smoothspeed* serveix per seguir el *target* de forma més fluida, creant un retard entre el moment en que el *target* es mou i el moment en que la càmera el comença a seguir. L'*Offset* serveix per col·locar la càmera a la posició desitjada respecte el *target*.

```
private void FixedUpdate()
{
    Vector3 desiredPosition = target.position + offset + displacement;
    Vector3 smoothPosition = Vector3.Lerp(transform.position,
desiredPosition, smoothspeed * Time.deltaTime);
    transform.position = smoothPosition;
}
```

7.4 IMPLEMENTACIÓ DELS SÍMPTOMES

A continuació s'explicarà breument la implementació de cada un dels símptomes programats en aquest prototip.

ESPASTICITAT

L'espasticitat es representa amb un número que va del 0 a l'1, sent 0 la no aparició del símptome, i 1 no poder-se moure. Quan aquest símptome apareix de forma aleatòria, l'espasticitat puja de 0 a 0.5. A l'script del control de personatge, *PlayerController*, la velocitat de moviment (*MaxSpeed*) i la força de salt (*JumpTakeOffSpeed*) són multiplicats per $(1 - \text{espasticitat})$.

```

/**SPASTICITY***/
if(OnSpasticity)
{
    if (frames >= random1 && spasticity == 0f)
    {
        spasticity = 0.5f;
        spasticated = true;
    }

    if(recoverSpasticity && spasticity == 0.5f)
    {
        spasticity = 0f;
        spasticated = false;
        random1 = Random.Range(frames + 1000, frames + 2000);
        recoverSpasticity = false;
    }
}

```

FATIGA

Per aquest prototip, en aquest símptome s'aprofita la variable d'espasticitat mencionada anteriorment per a representar la dificultat de moviment. Es representa com una barra d'energia que comença a 100 unitats. A mesura que es mou o es salta, aquesta energia es va gastant. Quan arriba a 50, l'espasticitat puja a 0.2. Quan baixa fins a 10, puja l'espasticitat fins a 0.8. Finalment a 0 d'energia, l'espasticitat està a 1, impeding qualsevol moviment.

```

/**FATIGUE***/
if (OnFatigue)
{
    if (FM.FatigueBar <= 50 && FM.FatigueBar > 10)
    {
        if (spasticated)
            spasticity = 0.5f;
        else
            spasticity = 0.2f;
    }
    else if (FM.FatigueBar <= 10 && FM.FatigueBar > 0)
        spasticity = 0.8f;
    else if (FM.FatigueBar == 0)
        spasticity = 1.0f;
    else
    {
        if (spasticated)
            spasticity = 0.5f;
        else
            spasticity = 0.0f;
    }
}

```

```

public void AddFatigue(float fatigue)
{
    if(FatigueBar >= 0)
    {
        if(FatigueBar < 100 && FatigueBar > 50)
        {
            GetComponent<Image>().color = green;
        }
        else if(FatigueBar <= 50 && FatigueBar > 10)
        {
            GetComponent<Image>().color = yellow;
        }
        else if(FatigueBar <= 10)
        {
            GetComponent<Image>().color = red;
        }
        FatigueBar -= fatigue;
        if (FatigueBar < 0)
            FatigueBar = 0;
        else if (FatigueBar > 100)
            FatigueBar = 100;

        Vector3 scale = transform.localScale;
        transform.localScale = new Vector3((FatigueBar / 100) * scaleX,
scale.y, scale.z);
    }
}

```

DESEQUILIBRIS

De forma semblant a la fatiga, els desequilibris es representen amb una barra que va de 0 a 100, amb la diferència que aquesta s'emplena mentre es desplaça i salta, en comptes de buidar-se. Quan arriba a 100 unitats (màxim), el personatge cau al terra i el jugador haurà d'utilitzar el botó de salt per reincorporar-se, ja que des del terra no es podrà moure.

```

public void AplyBalance(float balance)
{
    if (balance > 0)
    {
        if (balance > 100)
            balance = 100;

        Vector3 scale = transform.localScale;

        transform.localScale = new Vector3((balance / 100) * scaleX, scale.y,
scale.z);
    }
}

```

```

/**IMBALANCE***/
if (OnImbalance)
{
    float rand = 0;
    if (frames >= random3)
    {
        if (balancing)
            rand = Random.Range(-2, 2);
        else
            rand = Random.Range(-10, 10);
    }

    float fakeImbalance = imbalanceRange + rand;
    if (fakeImbalance > 99)
        fakeImbalance = 99;

    if(fakeImbalance >= 85 && fakeImbalance < 100)
        doubleVision.GetComponent<Animator>().SetBool("Tremble", true);
    else
        doubleVision.GetComponent<Animator>().SetBool("Tremble", false);

    if (imbalanceRange >= 100)
    {
        imbalanceRange = 100;
        imbalanced = true;
    }

    rotation = transform.localRotation;
    if (imbalanced && rotation.z >= -0.6f)
    {
        transform.localRotation = new Quaternion(rotation.x, rotation.y,
rotation.z - 0.01f, rotation.w);
    }

    if (balancing)
    {
        if (imbalanceRange > 10)
        {
            imbalanceRange -= 0.1f;
        }
        if (imbalanced)
            Balance();
    }

    BM.AplyBalance(fakeImbalance);
    random3 = Random.Range(frames + 500, frames + 800);
}

```

VISIÓ DOBLE

Aquest ha sigut el símptome més complexe d'implementar. Per aconseguir l'efecte de la visió doble, s'han afegit dos càmeres que es troben en la mateixa posició que la càmera principal. Aquestes càmeres són les anomenades *CameraRender*. En una carpeta del projecte, es van crear dues *Render Textures*, i a cada *CameraRender* se li va assignar una a la variable *Target Texture* de l'inspector. Llavors al Canvas, es van posar dues imatges que ocupessin la totalitat de la pantalla. A cada una d'aquestes imatges, se li va assignar la *Render Texture* de cada càmera. D'aquesta forma, quan es vol activar la visió

doble, només cal canviar, des de l'script, l'Offset (mencionat al 7.3) de les dues càmeres per a que quedin separades.

```
public void Activate(float displacement)
{
    if (displacement != 0)
    {
        Cam1.displacement = new Vector3(displacement, 0, 0);
        Visions[0].GetComponent<RawImage>().color = new Color(1, 1, 1, 0.5f);
        Cam2.displacement = new Vector3(-displacement, 0, 0);
        Visions[1].GetComponent<RawImage>().color = new Color(1, 1, 1, 0.5f);

        Visions[2].SetActive(true);

        //active = true;
        MainCamera.transform.rotation = new Quaternion(-180, 0, 0, 0);
    }
    else
    {
        Cam1.displacement = Vector3.zero;
        Visions[0].GetComponent<RawImage>().color = new Color(1, 1, 1, 1);
        Cam2.displacement = Vector3.zero;
        Visions[1].GetComponent<RawImage>().color = new Color(1, 1, 1, 1);

        Visions[2].SetActive(false);

        //active = false;
        MainCamera.transform.rotation = new Quaternion(0, 0, 0, 0);
    }
}
```

8 RESULTATS

En aquest apartat es miraran els objectius que es van marcar en un principi i llavors es farà la valoració de si s'han assolit i amb quin grau de compliment.

Es recorda que l'objectiu principal era dissenyar un videojoc, amb la condició de simular un desenvolupament professional. L'altra condició més important que marcava el marc del projecte era que havia de posar al jugador a la pell d'un malalt d'Esclerosis Múltiple. Finalment es volia fer una versió prototip per mostrar el concepte en execució.

Per a tals qüestions, es va decidir marcar uns objectius específics:

- Estudiar el desenvolupament i els símptomes de L'Esclerosis Múltiple.

Durant el transcurs del desenvolupament d'aquest projecte, sobretot en els primers passos de disseny i estructuració del joc, es va fer un estudi de la malaltia per a poder recrear-la amb fidelitat en el prototip final. Van haver-hi vèries reunions amb neuròlegs de l'hospital de Sta. Caterina, així com infermers especialitzats en l'EM i una entrevista amb una pacient diagnosticada. A més, es van fer lectures del llibre *Manejo Del Paciente Con Esclerosis Múltiple En Atención Primaria*, llibre otorgat pel Dr. Lluís Ramió. També es van fer consultes a internet sobre la malaltia i els seus símptomes.

- Implementar els controls bàsics d'un joc de plataformes 2D.

Pel prototip final, s'han implementat les mecàniques de desplaçar-se i de saltar, a més d'una càmera que segueix al jugador durant tota l'acció de la partida, pel qual s'obté una experiència bàsica per un plataformes 2D, tal i com es buscava.

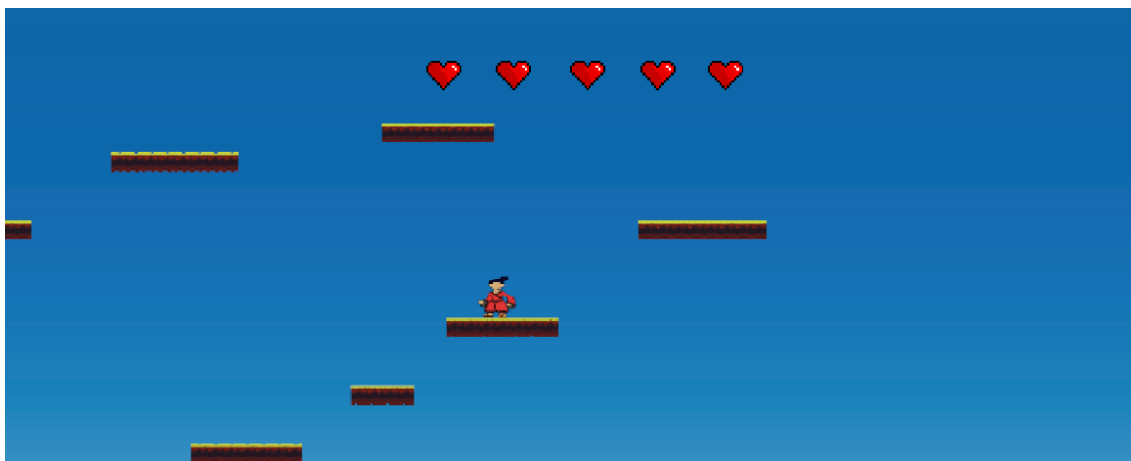


Figura 51 Nivell amb mecàniques bàsiques

No s'ha aprofundit excessivament en els controls bàsics, ja que s'hi ha dedicat més esforç en l'implementació dels símptomes de l'Esclerosis Múltiple.

- Implementar una representació dels diferents símptomes de la malaltia.

S'han implementat fins a 4 símptomes de la patologia, incloent la fatiga, l'espasticitat, els desequilibris i la visió doble. Es va dissenyar la seva representació en el videojoc quan es van fer les reunions i entrevistes, es van implementar i finalment es van fer els canvis

necessaris que se'ns va recomenar una vegada feta la reunió de seguiment en mig del desenvolupament.

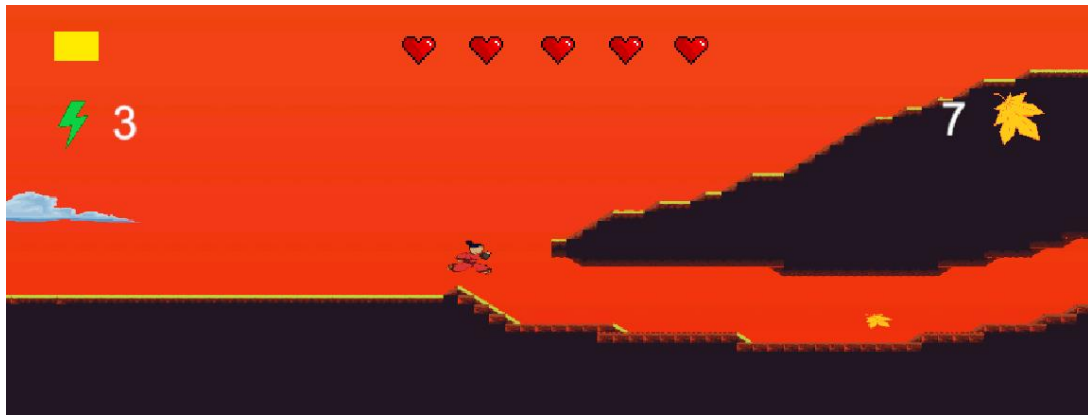


Figura 52 Amb poca energia, el jugador no pot arribar a saltar aquest obstacle.

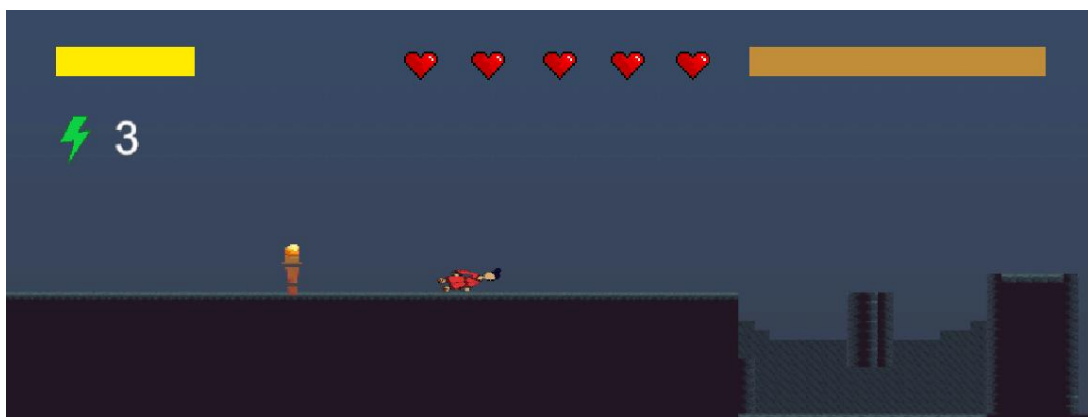


Figura 53 Protagonista caiguda al terra degut als desequilibris.

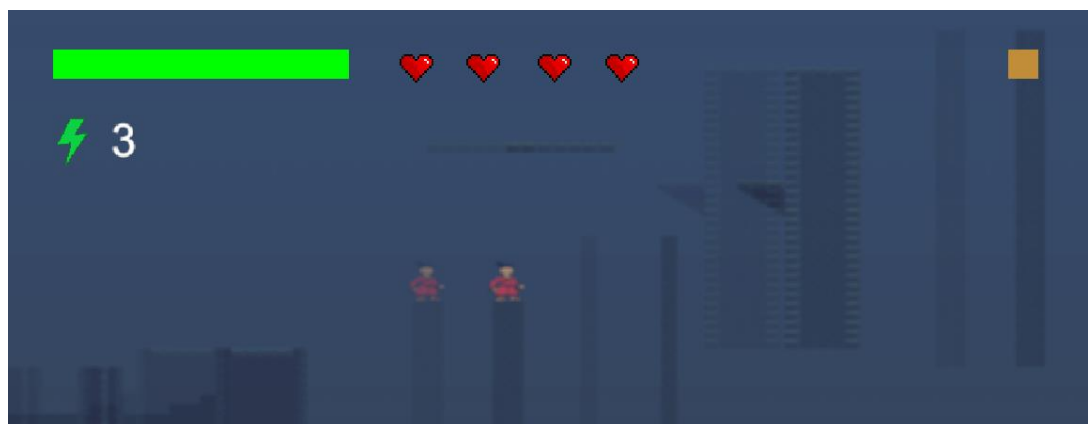


Figura 54 Amb la visió doble, fins que el jugador no es "tapi un ull", serà més difícil saltar aquest obstacle.

- Dissenyar nivells adaptats a les mecàniques seleccionades

En el prototip treballat s'han dissenyat tres nivells. Cada un d'aquests nivells representa una porció d'un nivell hipotètic més gran i complet. Aquests nivells s'han dissenyat tenint en compte les mecàniques i els símptomes, de manera que el jugador hagi de gestionar l'energia per no tenir fatiga, haurà de recol·lectar fulles (recuperadors d'espasticitat) repartits per tot el nivell per lluitar contra l'espasticitat, haurà de mantenir l'equilibri per no caure al buit i haurà de "tapar-se un ull" si no vol que la visió doble el fagi saltar en la plataforma equivocada.

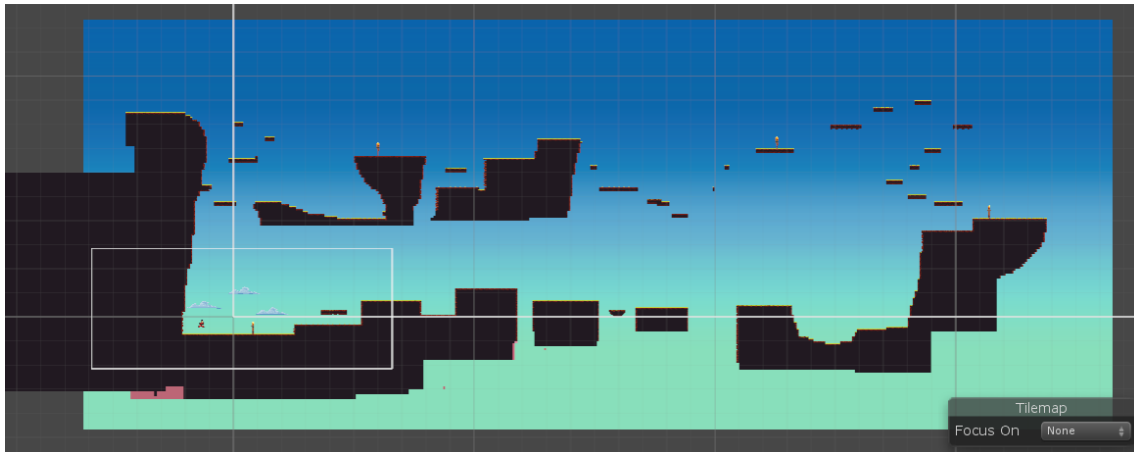


Figura 55 Disseny del primer nivell.

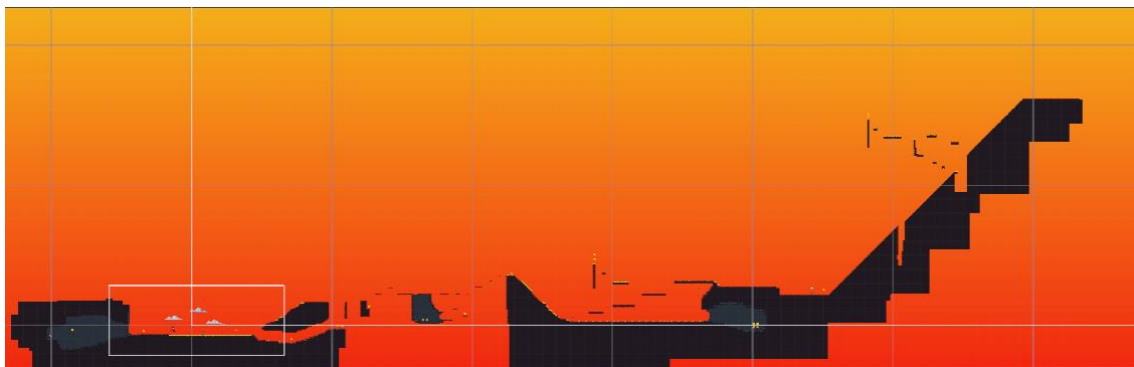


Figura 56 Disseny del segon nivell, on la protagonista pateix de fatiga i espasticitat.

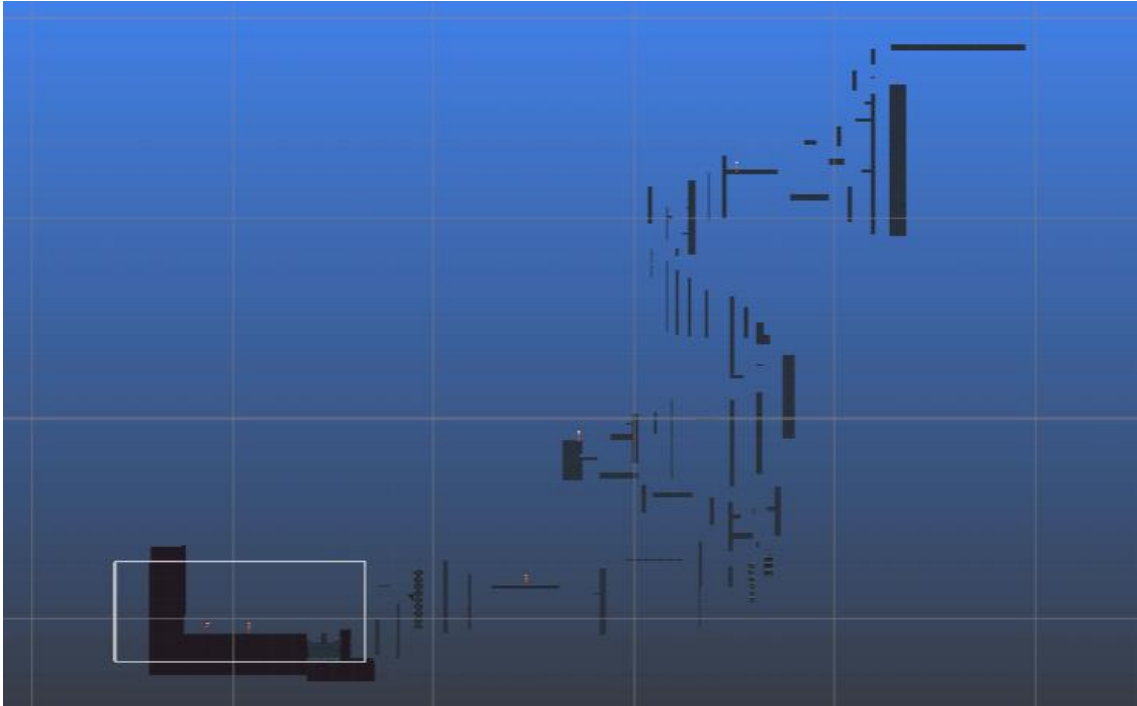


Figura 57 Disseny del tercer nivell, on la protagonista patirà de fatiga, desequilibris i visió doble.

9 CONCLUSIONS

Amb aquest treball s'ha pogut posar en pràctica tot l'après durant el grau de disseny i desenvolupament de videojocs, i s'ha pogut veure que encara hi ha moltes coses per aprendre.

Quan va sorgir la idea per primer cop per motius personals, no era gaire clar cap a quin camí es dirigiria, quina estructura tindria o quines mecàniques hi hauria. Una cosa era clara, doncs era voler visibilitzar una malaltia i, en certa manera, desenvolupar un joc que anés al revés de les convencions del disseny de videojocs; un disseny que no volgués proporcionar una fantasia de poder al jugador donant cada vegada més habilitats i recursos, sinó que les noves mecàniques que apareguessin creessin noves dificultats.

Finalment es va decidir fer un joc de plataformes 2D i adaptar la representació dels símptomes amb els reptes del gènere. Les reunions i entrevistes amb neuròlegs i pacients van ser una molt bona experiència i van ser molt útils per a una millor estructuració i un desenvolupament més fidel. L'etapa de disseny, reunions i planificació va ser, llavors, la més interessant i enriquidora.

Una vegada començada la implementació, queda clara la quantitat de feina que hi ha en el desenvolupament d'un videojoc i perquè les empreses de la indústria solen necessitar tant de personal, recursos i diners. Es van haver de fer moltes tasques d'àmbits diferents: programació de les mecàniques i el funcionament del joc, disseny i realització d'art i animacions, disseny de nivells. En aquest projecte, l'àmbit amb més pes ha sigut la programació dels símptomes.

Tot i que queden moltes idees que no han arribat al prototip per falta de temps o recursos, la finalització d'aquest projecte proporciona una recompensa molt satisfactòria. Desenvolupar un videojoc, tot i que sigui un prototip, comporta molta feina, però enllestir-lo és molt plaent.

Durant el transcurs de la implementació hi ha hagut numerables problemes en la programació, però Unity ofereix un manual molt ampli sobre com funcionen totes les seves funcions, a part que a internet es troben molts fòrums on la gent explica problemes que han tingut i com s'han solucionat.

El disseny del joc conté tot el que en un principi s'ha pensat, i el prototip té les parts més importants i amb les quals es pot entendre la idea del joc. En aquest prototip es poden veure tots els símptomes que es van voler implementar quan es va dissenyar el joc, a més de controlar una protagonista dissenyada i animada per l'autor de l'obra. L'objectiu del videojoc era visibilitzar una malaltia i posar al jugador a la pell d'una persona diagnosticada; la finalització d'aquest treball ha demostrat que s'ha pogut aconseguir.

10 TREBALL FUTUR

En aquest apartat es parlarà sobre noves característiques o idees que es podrien incorporar en el videojoc en un futur.

- Aquest prototip consta de tres nivells, cada un dels quals representa una porció d'un nivell més ampli i complet. No només en un hipotètic treball futur es farien els nivells complets, sinó que se'n farien de nous, incrementant la dificultat i la complexitat i afegint nous tipus de reptes.
- S'estudiarien nous símptomes per a poder representar més brots en nivells nous.
- Nivells interludi: entre nivell i nivell, hi hauria una petita secció on es controlaria a la protagonista a casa seva, just abans de partir cap a la peregrinació. En aquestes seccions es podrien representar les dificultats de la malaltia en un entorn més familiar i personal, com per exemple la forma que té d'actuar la fatiga quan comença el dia, la dificultat per agafar un objecte que està a dalt de tot d'una estanteria, o portar un objecte que es caigués de la mà per culpa de la movilitat dels dits.
- Un aprofundiment en la història podria facilitar una empatia cap a la protagonista i, per tant, una visibilització més personal.
- Tot l'art del videojoc podria acabar sent fet per l'autor de l'obra, ja sigui amb noves animacions, nous icones i escenaris propis.

11 BIBLIOGRAFIA

1. Manejo Del Paciente Con Esclerosis Múltiple En Atención Primaria: Diagnostico, Pronóstico Y Seguimiento.
 - a. Varis Autors
 - b. Realitzat per Luzan 5: <https://luzan5.com/>
 - c. 2017

2. Síntomes de l'esclerosi múltiple
 - a. Fundació Esclerosi Múltiple
 - b. 2019
 - c. https://www.fem.es/esclerosi-multiple/sintomes-de-lesclerosi-multiple/?gclid=EAlaIQobChMIlozL7O_H6wIVhoxRCh31RQGOEAAAYiAAEgK12_D_BwE

3. Esclerosis Múltiple
 - a. Mayo Clinic
 - b. 2020
 - c. <https://www.mayoclinic.org/es-es/diseases-conditions/multiple-sclerosis/symptoms-causes/syc-20350269>

4. Unity 2D
 - a. Unity Technologies
 - b. 09/01/2020
 - c. <https://docs.unity3d.com/Manual/Unity2D.html>

5. Brackeys
 - a. Youtuber
 - b. <https://www.youtube.com/user/Brackeys>

6. 2D Platformer Controller
 - a. Matt Shell
 - b. 30/04/2019
 - c. <https://learn.unity.com/tutorial/live-session-2d-platformer-character-controller#5c7f8528edbc2a002053b68e>

7. Disseny Conceptual de Videojocs
 - a. Imma Boada i Núria Puig
 - b. Assignatura de GDDV
 - c. Universitat de Girona

12 ANNEXOS

Aquí es recopilen altres scripts realitzats que no s'han mencionat en l'apartat d'Implementació:

Blur.shader

```
// Upgrade NOTE: replaced 'mul(UNITY_MATRIX_MVP,*)' with 'UnityObjectToClipPos(*)'

Shader "Custom/BLUR" {
    Properties{
        _Color("Main Color", Color) = (1,1,1,1)
        _BumpAmt("Distortion", Range(0,128)) = 10
        _MainTex("Tint Color (RGB)", 2D) = "white" {}
        _BumpMap("Normalmap", 2D) = "bump" {}
        _Size("Size", Range(0, 20)) = 1
    }

    Category{

        Tags { "Queue" = "Transparent" "IgnoreProjector" = "True"
"RenderType" = "Opaque" }

        SubShader {

            GrabPass {
                Tags { "LightMode" = "Always" }
            }
            Pass {
                Tags { "LightMode" = "Always" }

                CGPROGRAM
                #pragma vertex vert
                #pragma fragment frag
                #pragma fragmentoption ARB_precision_hint_fastest
                #include "UnityCG.cginc"

                struct appdata_t {
                    float4 vertex : POSITION;
                    float2 texcoord: TEXCOORD0;
                };

                struct v2f {
                    float4 vertex : POSITION;
                    float4 uvgrab : TEXCOORD0;
                };

                v2f vert(appdata_t v) {
                    v2f o;
                    o.vertex = UnityObjectToClipPos(v.vertex);
                    #if UNITY_UV_STARTS_AT_TOP
                    float scale = -1.0;
                    #else
                    float scale = 1.0;
                    #endif
                    o.uvgrab.xy = (float2(o.vertex.x,
o.vertex.y * scale) + o.vertex.w) * 0.5;
                    o.uvgrab.zw = o.vertex.zw;
                    return o;
                }

                sampler2D _GrabTexture;
                float4 _GrabTexture_TexelSize;
                float _Size;
            }
        }
    }
}
```

```

half4 frag(v2f i) : COLOR {
    half4 sum = half4(0,0,0,0);
    #define GRABPIXEL(weight,kernelx)
    tex2Dproj( _GrabTexture, UNITY_PROJ_COORD(float4(i.uvgrab.x +
    _GrabTexture_TexelSize.x * kernelx*_Size, i.uvgrab.y, i.uvgrab.z, i.uvgrab.w))) *
    weight
        sum += GRABPIXEL(0.05, -4.0);
        sum += GRABPIXEL(0.09, -3.0);
        sum += GRABPIXEL(0.12, -2.0);
        sum += GRABPIXEL(0.15, -1.0);
        sum += GRABPIXEL(0.18, 0.0);
        sum += GRABPIXEL(0.15, +1.0);
        sum += GRABPIXEL(0.12, +2.0);
        sum += GRABPIXEL(0.09, +3.0);
        sum += GRABPIXEL(0.05, +4.0);

    return sum;
}
ENDCG
}

GrabPass {
    Tags { "LightMode" = "Always" }
}
Pass {
    Tags { "LightMode" = "Always" }

CGPROGRAM
#pragma vertex vert
#pragma fragment frag
#pragma fragmentoption
ARB_precision_hint_fastest

#include "UnityCG.cginc"

struct appdata_t {
    float4 vertex : POSITION;
    float2 texcoord: TEXCOORD0;
};

struct v2f {
    float4 vertex : POSITION;
    float4 uvgrab : TEXCOORD0;
};

v2f vert(appdata_t v) {
    v2f o;
    o.vertex =
    UnityObjectToClipPos(v.vertex);

    #if UNITY_UV_STARTS_AT_TOP
    float scale = -1.0;
    #else
    float scale = 1.0;
    #endif
    o.uvgrab.xy = (float2(o.vertex.x,
    o.vertex.y * scale) + o.vertex.w) * 0.5;
    o.uvgrab.zw = o.vertex.zw;
    return o;
}

sampler2D _GrabTexture;
float4 _GrabTexture_TexelSize;

```

```

float _Size;

half4 frag(v2f i) : COLOR {
    half4 sum = half4(0,0,0,0);
    #define GRABPIXEL(weight,kernely)
tex2Dproj( _GrabTexture, UNITY_PROJ_COORD(float4(i.uvgrab.x, i.uvgrab.y +
_GrabTexture_TexelSize.y * kernely*_Size, i.uvgrab.z, i.uvgrab.w))) * weight

    sum += GRABPIXEL(0.05, -4.0);
    sum += GRABPIXEL(0.09, -3.0);
    sum += GRABPIXEL(0.12, -2.0);
    sum += GRABPIXEL(0.15, -1.0);
    sum += GRABPIXEL(0.18, 0.0);
    sum += GRABPIXEL(0.15, +1.0);
    sum += GRABPIXEL(0.12, +2.0);
    sum += GRABPIXEL(0.09, +3.0);
    sum += GRABPIXEL(0.05, +4.0);

    return sum;
}
ENDCG
}

GrabPass {
    Tags { "LightMode" = "Always" }
}
Pass {
    Tags { "LightMode" = "Always" }

CGPROGRAM
#pragma vertex vert
#pragma fragment frag
#pragma fragmentoption

ARB_precision_hint_fastest

#include "UnityCG.cginc"

struct appdata_t {
    float4 vertex : POSITION;
    float2 texcoord: TEXCOORD0;
};

struct v2f {
    float4 vertex : POSITION;
    float4 uvgrab : TEXCOORD0;
    float2 uvbump : TEXCOORD1;
    float2 uvmain : TEXCOORD2;
};

float _BumpAmt;
float4 _BumpMap_ST;
float4 _MainTex_ST;

v2f vert(appdata_t v) {
    v2f o;
    o.vertex =

UnityObjectToClipPos(v.vertex);

    #if UNITY_UV_STARTS_AT_TOP
    float scale = -1.0;
    #else
    float scale = 1.0;
    #endif

```


EnergyCounter.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class EnergyCounter : MonoBehaviour
{
    public PlayerController pc;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        GetComponent<Text>().text = pc.maxEnergy.ToString();
    }
}
```

LeafCounter.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class LeafCounter : MonoBehaviour
{
    public PlayerController pc;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        GetComponent<Text>().text = pc.numLeafs.ToString();
    }
}
```

LivesCounter.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class LivesCounter : MonoBehaviour
{
    public float index;

    public PlayerController pc;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (pc.numLives < index)
            GetComponent<Image>().enabled = false;
        else
            GetComponent<Image>().enabled = true;
    }
}
```

PhysicsObject.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PhysicsObject : MonoBehaviour
{
    public float minGroundNormalY = .65f;
    public float gravityModifier = 1f;

    protected Vector2 targetVelocity;
    protected bool grounded;
    protected Vector2 groundNormal;
    protected Rigidbody2D rb2d;
    protected Vector2 velocity;
    protected ContactFilter2D contactFilter;
    protected RaycastHit2D[] hitBuffer = new RaycastHit2D[16];
    protected List<RaycastHit2D> hitBufferList = new List<RaycastHit2D>(16);

    protected const float minMoveDistance = 0.001f;
    protected const float shellRadius = 0.01f;

    void OnEnable()
    {
        rb2d = GetComponent<Rigidbody2D>();
    }

    void Start()
    {
        contactFilter.useTriggers = false;
        contactFilter.SetLayerMask(Physics2D.GetLayerCollisionMask(gameObject.layer));
        contactFilter.useLayerMask = true;
    }

    void Update()
    {
        targetVelocity = Vector2.zero;
        ComputeVelocity();
        ComputeSymptoms();
    }

    protected virtual void ComputeVelocity()
    {
    }

    protected virtual void ComputeSymptoms()
    {
    }

    void FixedUpdate()
    {
        velocity += gravityModifier * Physics2D.gravity * Time.deltaTime;
        velocity.x = targetVelocity.x;

        grounded = false;

        Vector2 deltaPosition = velocity * Time.deltaTime;
    }
}

```

```

Vector2 moveAlongGround = new Vector2(groundNormal.y, -groundNormal.x);

Vector2 move = moveAlongGround * deltaPosition.x;

Movement(move, false);

move = Vector2.up * deltaPosition.y;

Movement(move, true);
}

void Movement(Vector2 move, bool yMovement)
{
    float distance = move.magnitude;

    if (distance > minMoveDistance)
    {
        int count = rb2d.Cast(move, contactFilter, hitBuffer, distance +
shellRadius);
        hitBufferList.Clear();
        for (int i = 0; i < count; i++)
        {
            hitBufferList.Add(hitBuffer[i]);
        }

        for (int i = 0; i < hitBufferList.Count; i++)
        {
            Vector2 currentNormal = hitBufferList[i].normal;
            if (currentNormal.y > minGroundNormalY)
            {
                grounded = true;
                if (yMovement)
                {
                    groundNormal = currentNormal;
                    currentNormal.x = 0;
                }
            }

            float projection = Vector2.Dot(velocity, currentNormal);
            if (projection < 0)
            {
                velocity = velocity - projection * currentNormal;
            }

            float modifiedDistance = hitBufferList[i].distance - shellRadius;
            distance = modifiedDistance < distance ? modifiedDistance : distance;
        }
    }

    rb2d.position = rb2d.position + move.normalized * distance;
}
}

```

13 MANUAL D'INSTAL·LACIÓ I D'USUARI

13.1 MANUAL D'INSTAL·LACIÓ

En aquest apartat s'explicarà com instal·lar el videojoc treballat per un ordinador amb Windows.

1. Primer de tot, es necessitarà descarregar l'arxiu comprimit on es trobarà l'aplicació.



Figura 58 Arxiu comprimit del projecte.

2. Seguidament, es fa clic dret i es busca l'opció de descomprimir.

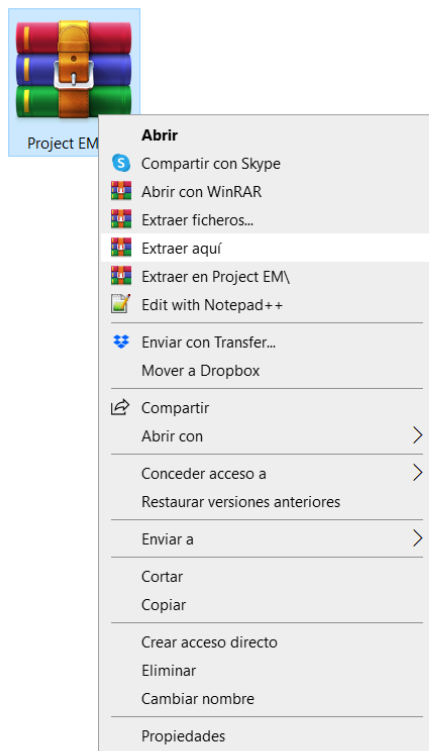


Figura 59 Descompriment de l'arxiu.

3. Finalment es podrà entrar a la carpeta extreta i allà es trobarà l'executable, anomenat *Project EM.exe*. Al fer doble clic se'ns obrirà el joc.

Nombre	Fecha de modificación	Tipo
MonoBleedingEdge	01/09/2020 23:19	Carpeta de archivos
Project EM_Data	02/09/2020 1:12	Carpeta de archivos
Project EM.exe	26/09/2019 21:01	Aplicación
UnityCrashHandler32.exe	26/09/2019 20:57	Aplicación
UnityPlayer.dll	26/09/2019 21:03	Extensión de la ap...

Figura 60 Executable del joc.

13.2 MANUAL D'USUARI

A continuació, s'adjunta en forma d'apartat un petit manual d'usuari mostrant el funcionament de l'aplicació prototip i detallant les accions que es poden dur a terme. Primer, en executar el videojoc, apareixerà una pantalla amb el logotip de Unity (*splash screen*), la qual és obligatòria si s'empra el motor en la seva versió gratuïta (Figura 61).



Figura 61 Splash Screen de Unity.

Seguidament, s'entra en el menú principal del videojoc. En aquesta pantalla no es pot utilitzar el mando de consola i s'haurà d'interactuar amb el ratolí. Les dues opcions disponibles en aquest menú són començar el joc (Start) o sortir i tancar l'aplicació (Quit).

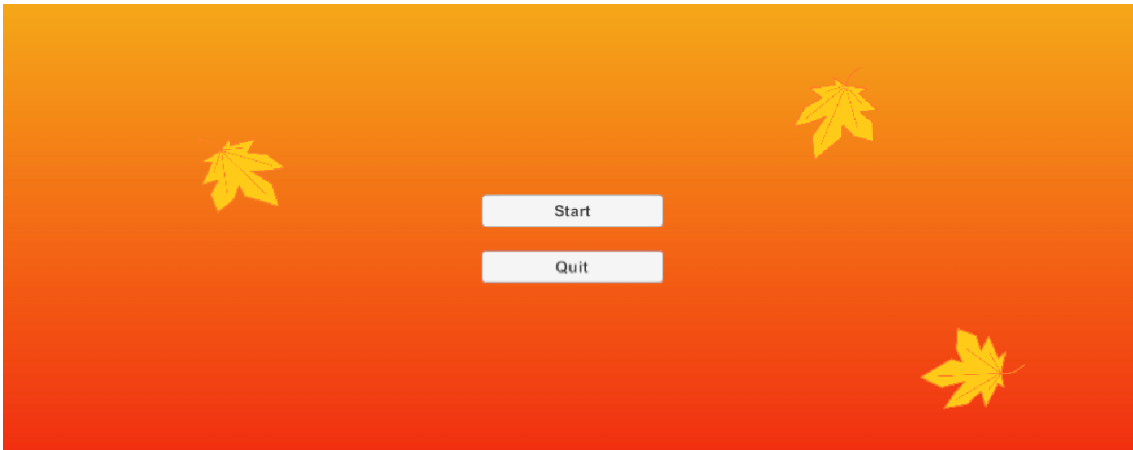


Figura 62 Menú Principal del joc.

En fer clic a Start, es començarà a jugar i es portarà directament al primer nivell. En aquest nivell la protagonista encara no ha fet cap brot de la malaltia, ja que serveix com a tutorial pels controls bàsics del joc: moure's i saltar.

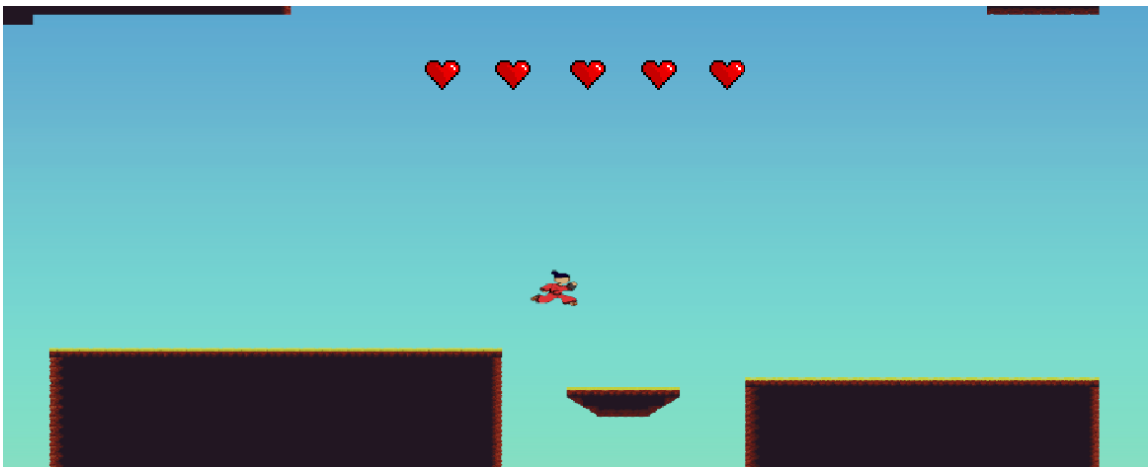


Figura 63 Primer Nivell

Quan s'arriba al final d'un nivell, es transporta directament al següent, i així successivament fins que el joc acaba i es porta de nou al menú principal.

En el segon nivell, la protagonista pateix de dos símptomes: la fatiga i l'espasticitat. Ja que l'espasticitat requereix de fulles per combatre el símptoma, en tot el nivell hi ha distribuïts fulles per a recol·lectar. És un nivell amb petites zones i plataformes opcionals on la recompensa de la seva exploració són fulles extremes.

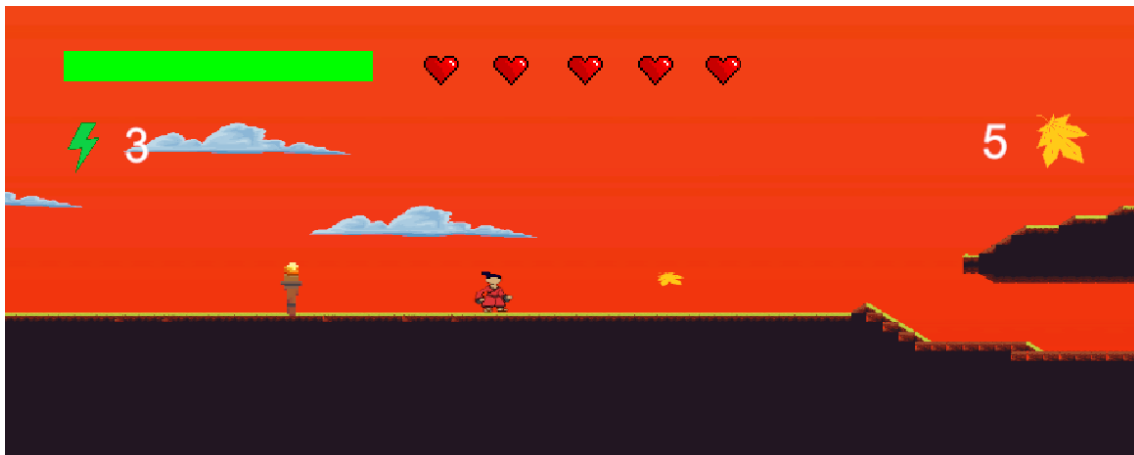


Figura 64 Segon Nivell

Finalment, en el 3r nivell, situat en un altre brot, la protagonista pateix de desequilibris, visió doble i fatiga, aquest últim es repeteix ja que és un dels símptomes més freqüents en tota la malaltia i la més present en el dia a dia.

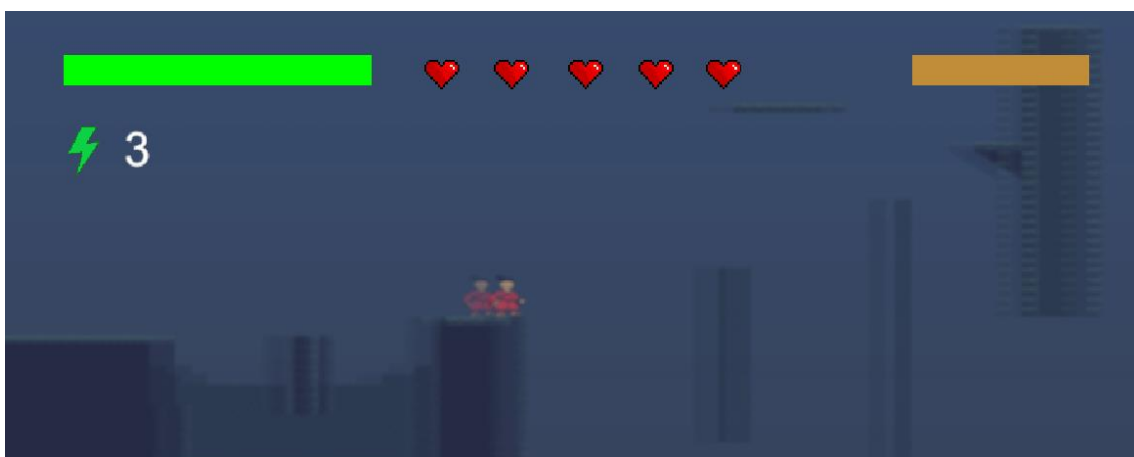


Figura 65 Tercer Nivell

En cada un dels nivells, el jugador té l'opció d'anar al menú d'opcions. Des d'aquest menú, es podrà retornar al menú principal o continuar a la partida des d'el moment que s'havia deixat.



Figura 66 Menú d'Opcions

Una vegada acabat el tercer nivell, es tornarà directament al menú principal i, des d'allà, es podrà sortir del joc prement el botó *Quit*.

