

TREBALL DE RECERCA

White Caves

Disseny i desenvolupament d'un videojoc

Recerca bibliogràfica sobre els efectes
psicològics dels videojocs



MARÇAL PONS WENDENBURG

2n BATXILLERAT A

TUTOR: ALEXANDRE ALFARO MAGALDI
INSTITUT SANTIAGO SOBREQUÈS I VIDAL

9 DE NOVEMBRE DEL 2020

“...el joc és més vell que la cultura, perquè aquesta sempre pressuposa una societat humana i els animals no s’han esperat que l’home els ensenyés a jugar.”

Johan Huizinga, 1938

*Homo ludens, proeve eener bepaling van
het spel-element der cultuur*

Agraïments

En primer lloc, vull reconèixer el meu tutor del Treball de Recerca Alexandro Alfaro per ajudar a reencaminar-me quan estava perdut i donar-me suport i orientació en tot el que he necessitat.

També vull donar les gràcies a totes aquelles persones que m'han ajudat a crear un millor videojoc. En Joan Aparicio de SandBloom Studio em va concedir una entrevista i va redactar molt bones respostes. Els amics m'han aconsellat i fet crítiques constructives, especialment l'Íñu Montero i en Néstor Gargallo. La Mar Álvarez va compondre la música del joc i en Darío Nuviola, l'Iris Marti, en Jaume Viñas i la Mar la van interpretar. Les comunitats d'usuaris de Discord de "Gabbitt 3D" i "Official Unity Discord" van respondre ràpidament les meves preguntes sobre modelatge 3D i programació. Els més de 50 jugadors de la versió Beta de White Caves han testat el joc i donat retroacció. Els creadors de contingut de YouTube i blocs en línia han creat explicacions i *tutorials* que m'han ensenyat a utilitzar els programes.

Així mateix, vull agrair el meu pare i la meva germana per recolzar-me quan em quedava encallat i ajudar-me a estructurar la memòria escrita i revisar-la.

Índex

1.	Introducció.....	3
1.1	Objectius	3
1.2	Metodologia	4
1.3	Els videojocs.....	6
1.3.1	Història dels videojocs	6
1.3.2	Principals gèneres de videojocs.....	8
2.	Efectes psicològics dels videojocs	10
2.1	Efectes positius.....	10
2.2	Efectes negatius.....	12
2.2.1	Violència.....	12
2.2.2	Altres efectes negatius	13
3.	Introducció als llenguatges de programació i als <i>Game Engines</i>	15
3.1	Llenguatges de programació.....	15
3.2	Motors de Joc	19
3.2.1	Gràfics.....	19
3.2.2	Àudio.....	19
3.2.3	Creació de xarxes	20
3.2.4	Físiques	20
3.2.5	<i>Graphical user interfases (GUI)</i>	20
3.2.6	Programació.....	21

4.	Realització d'un videojoc.....	22
4.1	Disseny.....	22
4.1.1	La història.....	22
4.1.2	Definir el món i els <i>items</i>	28
4.1.3	Dissenyar els personatges	33
4.1.4	So	35
4.2	Producció.....	37
4.2.1	Modelatge 3D i creació de textures.....	37
4.2.2	Animació 3D.....	46
4.2.3	Programació.....	48
4.2.4	Fase de proves	61
5.	Conclusió i valoració personal	63
	Referències.....	65
	Annex 1: Entrevista a Sand Bloom Studio	68
	Annex 2: Esbossos del disseny del món.....	72
	Dibuixos a llapis.....	72
	Dibuixos digitals.....	73
	Disseny final del món.....	74

1. Introducció

A 3r d'ESO vaig començar un treball de recerca sobre exoplanetes, amb la intenció de fer seguiment durant dos anys d'un d'aquests cossos celestes. Malauradament, després d'un any, únicament havia fet recerca bibliogràfica perquè havia estat difícil contactar amb l'observatori on s'havien d'obtenir les dades. Així doncs, a principis de 4t d'ESO vaig canviar el tema a "Reconstrucció virtual del castell de Sant Iscle", amb l'objectiu de fer un model 3D de l'estat actual d'aquest castell i un altre de la meva hipòtesi de com hauria estat a l'edat mitjana. Tanmateix, després d'un altre any de fer recerca bibliogràfica sobre la fortificació però no haver creat cap dels models 3D, em vaig desanimar per la manca d'aplicació pràctica, així que vaig tornar a canviar el tema.

A l'octubre del 2019 em va sorgir la idea de fer un videojoc pel treball de recerca. La intenció inicial era situar-lo al castell de Sant Iscle per aprofitar l'anterior treball inacabat, però més tard em vaig adonar que el disseny de la història no era adequat. En comptes d'arreglar-lo, em vaig començar un de nou, un joc plataformes, en tres dimensions i amb un important missatge darrere.

Els videojocs han crescut en popularitat exponencialment les últimes dècades. D'altra banda, pràcticament cap jugador sap realment com es crea un videojoc, sinó que es limita a gaudir-lo. Això és correcte i, de fet, és el seu objectiu. Malgrat això, la meua curiositat va fer que, cap al 2016, em comencés a interessar pel modelatge 3D i la programació. Primer, vaig intentar fer petits videojocs dins Minecraft. No en vaig acabar mai cap, però em van donar experiència sobre el disseny de jocs. Després, vaig decidir fer-ne un de veritat, seguint tutorials de YouTube, però van ser projectes molt efímers perquè ni tan sols sabia programar. Finalment, he pogut materialitzar aquestes idees durant el Treball de Recerca de Batxillerat.

1.1 Objectius

Aquest treball té per objectius:

- En primer lloc, aprendre a dissenyar i desenvolupar un videojoc i a utilitzar els programes necessaris.
- En segon lloc, aplicar aquests coneixements per a crear un videojoc i, finalment, publicar-lo a Internet i millorar-lo gràcies a les crítiques.
- En tercer lloc, aprendre com les empreses de videojocs inclouen una crítica social o un missatge en els seus títols, i aplicar aquests coneixements per tal que el meu videojoc contingui un missatge sobre la superació i millora personal.

- En quart lloc, aprendre els efectes psicològics dels videojocs, realitzant-ne una recerca bibliogràfica extensiva i sintetitzant-ne la informació.

1.2 Metodologia

Per dur a terme aquest treball, vaig començar buscant informació a Internet sobre com crear un videojoc. Des d'un inici vaig aprendre l'ús de programes de modelatge, animació 3D, creació de textures, maquetació i programació (Blender i Unity a partir de tutorials de Youtube i publicacions a les pàgines oficials; Visual Studio Code de forma autodidàctica) i a programar amb C# (amb els cursos gratuïts de CodeAcademy i SoloLearn i videocursos de YouTube de Brackeys i Code Monkey). Per aprendre a dissenyar una bona història i provocar sentiments en el jugador vaig mirar desenes de petits documentals per Internet, la majoria orientats als videojocs, però altres a la creació de guions cinematogràfics o d'art en general. També em vaig apuntar a un curs en línia de *Coursera* anomenat "Story and Narrative Development for Video Games" per aprendre a dissenyar una bona història, però no el vaig finalitzar, ja que eren coneixements que jo ja tenia assolits. Addicionalment, vaig realitzar un videocurs sobre desenvolupament de videojocs impartit per *FutureLearn*. Tot i que realment no em va donar noves eines específiques, em va aportar una idea general de com es crea un joc.

Mentrestant, vaig crear un petit videojoc en tres dimensions per familiaritzar-me amb els programes. Després, com a segon videojoc, vaig dissenyar una història situada en el castell de Sant Iscle. Malauradament, com s'explica a l'apartat 4.1.1, la història era difícil de convertir en un videojoc. Per tant, vaig tornar a començar de nou, seguint un mètode per dissenyar videojocs més pautat i utilitzant els coneixements apresos. En aquest segon intent vaig tenir en compte les crítiques que m'havien dit amics coneixedors del tema. Tant la creació dels models 3D com la programació del joc van ser processos molt divertits i reconfortants la majoria del temps, però alhora desmotivants i frustrants en certes ocasions. No obstant, gràcies a Internet i a fòrums en línia, vaig poder obtenir ràpidament la solució a la majoria de problemes amb què em topava.

Per a poder aprendre com empreses que creen videojocs inclouen missatges en els seus títols vaig anar a l'exposició temporal *Gameplay* oferta pel Centre de Cultura Contemporània de Barcelona (CCCB). Així mateix, també vaig fer una entrevista telemàtica a l'estudi de desenvolupament de videojocs *indie* Sand Bloom Studio. Les conclusions obtingudes em van servir per incorporar un missatge de superació en el videojoc.

Finalment, vaig fer una revisió bibliogràfica d'articles acadèmics sobre els efectes psicològics dels videojocs, utilitzant les paraules clau "videogame" i "psychological effects" en el cercador Google Scholar. Per fer el

resum que incloc en aquest treball m'he basat sobretot en l'article *The psychological effects of videogames on young people: a review*, escrit per Lavinia McLean i Mark Griffiths, i el bloc en línia *The Psychological Effects Of Video Games*, publicat a <https://medium.com>. Addicionalment, he llegit nombrosos resums d'articles per a aprofundir més en certs aspectes de la meva recerca.

1.3 Els Videojocs

1.3.1 Història dels videojocs

Els primers videojocs utilitzaven aparells electrònics interactius amb diferents formats de visualització. Podríem dir que el primer exemple es remunta al 1947 amb el joc creat per Thomas T. Goldsmith Jr. i Estle Ray Mann i anomenat *Cathode-ray tube amusement device* (o Aparell de diversió a partir d'un tub de raigs catòdics) [1]. Consisteix en un aparell analògic que permetia al jugador controlar un punt a la pantalla que simulava un míssil disparat a diverses dianes.

El 1958 William Higinbotham va crear *Tennis for Two* (vegeu Fig. 1) [1], joc que sobretot serví com a demostració científica. Més endavant, el 1972, Atari el llençaria sota el nom *Pong*, el primer joc *Arcade*.

Mentrestant, el 1962 es va llençar *Spacewar!*, considerat el primer videojoc d'ordinador que es va comercialitzar [2]. Fou el primer videojoc que es podia jugar alhora des de diferents instal·lacions computacionals.

El 1972 va entrar al mercat la *Magnavox Odyssey* (vegeu Fig. 2), la primera videoconsola de casa, creada per Ralph Baer. Generava formes a la pantalla que podien ser controlades i interactuar-hi. Era en blanc i negre i no tenia so.

Als anys vuitanta van aparèixer jocs molt exitosos com *Space Invaders*, *Tetris* o *Pac-Man*. El 1980, Nintendo va crear la sèrie de videoconsols portàtils *Game & Watch* (Joc i rellotge). El 1983 va presentar al Japó la videoconsola *Family Computer* (*FamiCom*), que a altres regions canviaria el nom per *Nintendo Entertainment System* (*NES*). Un any després Sega va treure al mercat la seva consola *Master System*, que competiria amb Nintendo pel lideratge. Aquesta competència indirectament fonamentaria la indústria del videojoc, que des d'aleshores només aniria en augment.

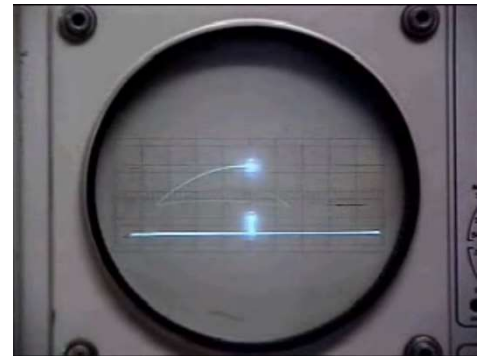


Fig. 1. **Tennis for Two**, un dels primers videojocs. La llum superior representa la pilota de tennis i la línia inferior el terra. Extret de: <https://youtu.be/s2E9iSQfGdg>.



Fig. 2. **Magnavox Odyssey**, la primera consola de casa. Extret de: https://es.wikipedia.org/wiki/Magnavox_Odyssey#/media/Archivo:Magnavox-Odyssey-Console-Set.png

El 1973 es va publicar el primer videojoc amb perspectiva 3D, *Maze War* (vegeu Fig. 3), que va inspirar a la majoria dels *First Person Shooter*¹ [2]. No obstant, l'escenari no estava compost d'objectes amb volum, sinó que eren una suma de fotogrames 2D que donaven al jugador l'efecte de tres dimensions.

Actualment, tot i que hi ha hagut molts avenços tecnològics, hi ha algunes companyies independents que segueixen apostant pels jocs 2D amb PixelArt, no només per nostàlgia al passat dels videojocs, sinó també com a recurs artístic. Tot i així, la majoria de videojocs aprofiten els nous recursos i opcions que els avenços en tecnologia han permès. Aquests són els més importants:

- Internet va permetre que diversos jugadors juguessin alhora des de pràcticament qualsevol lloc del món, creant comunitats de jugadors molt més grans i en molts casos amb més opcions de diversió.
- Els jocs per a dispositius mòbils han substituït en gran part les consoles portàtils i han augmentat notablement les vendes per a jocs per a passar l'estona.
- La realitat augmentada, la realitat virtual i els aparells que van enganxats al cos i que reproduïxen les sensacions que el personatge té perquè el jugador també les tingui (vegeu Fig. 4) han permès augmentar notablement el sentiment de formar part del joc.
- Nous programes han posat la creació de videojocs a l'abast de qualsevol.



Fig. 3. **Maze War**, primer videojoc amb perspectiva 3D. Extret de: https://www.youtube.com/watch?v=ZgT9IgrLrvI&ab_channel=brucedamer, CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=47507741>.



Fig. 4. **HardLight**, un equipament que vibra per a fer més real l'experiència, juntament amb un equipament convencional de realitat virtual (les ulleres i els controls). Extret de: <https://www.trendhunter.com/trends/>

¹ First Person Shooter: popular gènere de videojocs centrat en lluita generalment d'armes de foc i amb perspectiva en primera persona, és a dir, que la càmera se situa en la cara del personatge i el jugador veu el que veuria si realment estigués en aquell escenari.

1.3.2 Principals gèneres de videojocs

Els videojocs es classifiquen en gèneres, basant-se sobretot en les seves característiques o objectius i no tant en la seva narrativa[3]. La majoria de jocs es poden classificar en més d'una categoria, i les principals són [3]:

- Jocs d'acció:
 - Plataformes
 - *Shooters*
 - Lluita
 - Sigil
 - Supervivència
 - Rítmics
- Jocs d'acció-aventura
 - Supervivència terrorífica
 - Metroidvania
- Jocs d'aventures
 - Aventures textuais
 - Aventures gràfiques
 - Novel·les visuals
 - Pel·lícules interactives
- Jocs de rol (RPG)
 - RPG d'acció
 - *Massive multiplayer online role-playing games* (MMORPG)
 - RPG tàctics
 - RPG sandbox o de món obert
- Jocs de simulació
 - Simulació de construcció i gestió
 - Simulació de la vida real
 - Simulació de vehicles
- Jocs d'estratègia
 - 4X (*explore, expand, exploit and exterminate*)
 - Artilleria
 - *Real-time strategy* (RTS)
 - *Real-time tactics* (RTT)
 - *Multiplayer online battle arena* (MOBA)
 - *Tower defense*

- *Turn-based strategy* (TBS)
- *Turn-based t ctics* (TBT)
- Guerra
- Jocs d'esports
 - Curses
 - Esports d'equip
 - Competitiu
 - Lluita realista
- Jocs de puzles
 - L gica
 - Tr via
- Jocs desocupats
 - Desocupats
 - Casuals
 - Festius
 - De programaci 
 - Joc de taula
 - *Massive multiplayer online* (MMO)
 - Art stic
 - Educacional
 - *Exergame* (o joc d'esfor )

Aquest treball se centra sobretot en els jocs de plataformes, ja que  s la categoria que millor descriu White Caves. Aquesta categoria rep el nom, ja que el personatge interactua amb plataformes (normalment corrent, saltant o caient). Hi ha moltes classes de jocs de plataformes, i la majoria s n en 2D. El m s conegut  s *Super Mario Bros* (vegeu Fig. 5).



Fig. 5. **Super Mario Bros**, Captura de pantalla de la NES. Extret de: <https://gamefabrique.com/games/super-mario-bros/#screenshots>

2. Efectes psicològics dels videojocs

A causa del gran creixement de la popularitat dels videojocs ha sorgit un temor als seus possibles efectes negatius, sobretot dels jocs violents. Així doncs, en aquest apartat es du a terme una anàlisi de la literatura científica sobre aquest tema, basant-se sobretot en una revisió de Lavinia Griffiths i Mark McLean [4] i en una entrada de bloc [5].

2.1 Efectes positius

Per començar, diversos estudis han conclòs unànimement que els videojocs ajuden a millorar les habilitats perceptives, l'atenció visual i les habilitats espacials. Per exemple, un grup d'investigadors va concloure que tan sols una hora de jugar a videojocs d'acció és suficient per millorar l'atenció visual selectiva [6]. L'atenció visual selectiva és l'habilitat de mantenir-se concentrat malgrat les distraccions, i té sentit perquè molts videojocs tenen ambients commovedors i amb molts detalls on el jugador s'ha de centrar únicament en uns objectes o zones en concret. No obstant això, els efectes mesurats no són dràstics.

Els videojocs són grans eines de formació perquè es basen en un aprenentatge multisensorial, actiu, experimental i basat en problemes. En altres paraules, a diferència d'altres formes d'aprenentatge més tradicionals, els videojocs es basen en la motivació i implicació del jugador en la tasca i posen èmfasi en la pràctica distribuïda d'habilitats. De fet, actualment ja hi ha videojocs disponibles -centrats en l'aprenentatge basat en habilitats- que es consideren que són efectius a l'hora d'ensenyar gestió d'estratègies i assignatures tradicionals com àlgebra i biologia, i en millorar les habilitats informàtiques.

Diversos estudis han demostrat que els videojocs poden ajudar a mitigar o fins i tot revertir la pèrdua de memòria en gent gran. Per exemple, un estudi demostra com un videojoc en 3D específicament dissenyat per aquesta tasca pot millorar l'habilitat cognitiva de gent gran sana [7]. De fet, mostra que els voluntaris d'entre 60 i 85 anys que havien jugat 12 hores repartides en un mes van treure millor puntuació en les proves per mesurar l'habilitat cognitiva que un grup de la mateixa edat però que no havia estat entrenat i, fins i tot, que un grup d'entre 20 i 30 anys, que tampoc hi havia jugat mai. I aquesta millora en la capacitat cognitiva, bàsicament en la memòria operativa i atenció prolongada, no era momentània, sinó que es va mantenir durant sis mesos una vegada l'entrenament havia finalitzat.

També cal destacar els efectes positius dels videojocs seriosos com a eina per potenciar les conductes prosocials als jugadors, ja que aquests jocs fomenten els comportaments prosocials i d'ajuda als personatges per a poder progressar en el joc. Tot i que hi ha pocs jocs d'aquesta mena, i doncs, poca recerca que n'analitzi

els efectes, s'ha trobat una relació directa entre el fet de jugar prosocialment i un comportament prosocial a la vida real. Durant un estudi [8], es van assignar 191 nens d'entre 9 i 14 anys a un joc violent, un joc neutral o un joc prosocial. Malgrat certes limitacions, els investigadors van concloure que els jocs prosocials van augmentar el comportament d'ajuda a curt termini i els jocs violents van augmentar el comportament nociu i disminuir el positiu o innocu.

Els videojocs també han estat utilitzats com a forma de psicoteràpia (per exemple, un joc de realitat virtual on s'ha de pujar a una torre molt alta per a fer front a la por a les altures), per ajudar pacients amb càncer (per a distreure'ls dels tractaments invasius), com a teràpia ocupacional [9], com a teràpia física i en rehabilitació després d'una lesió traumàtica del cervell.

Un altre efecte psicològic positiu de certs videojocs és l'augment de la creativitat. En un estudi es va arribar a la conclusió que jugar certs videojocs pot augmentar la creativitat, però també és important com es juga. En aquest estudi es van dividir els 352 participants en quatre grups: el primer grup mirava la televisió; el segon grup jugava a un videojoc de curses; el tercer grup jugava a Minecraft² amb la instrucció de jugar "creativament"; i finalment el darrer grup també jugava a Minecraft però sense instruccions de com havien de jugar. Després de 40 minuts de joc o mirar la TV es van fer diversos tests per calcular la seva creativitat. El grup que va tenir més puntuació va ser el de jugadors de Minecraft que no havien rebut instrucció, però el que va obtenir-ne menys va ser el de jugadors d'aquest mateix joc que sí que havien rebut instrucció. Davant aquest segon resultat inesperat els investigadors van presentar les següents hipòtesis: donar-los la instrucció de ser creatius potser els va limitar a l'hora de jugar, resultant en una experiència menys creativa; potser van utilitzar tota la "capacitat creativa" en jugar i no els en va quedar a l'hora de fer la prova; o potser no els va agradar que els diguessin com jugar i van acabar rebutjant les instruccions. En tot cas, els investigadors van acabar conclouent que, tot i que hi ha videojocs que poden augmentar la creativitat, també és molt important com es juga, i que segurament Minecraft no és l'únic videojoc que produeix aquests efectes. Fins i tot, és possible trobar resultats semblants en jocs de lluita, ja que en ells s'ha de desenvolupar una estratègia.

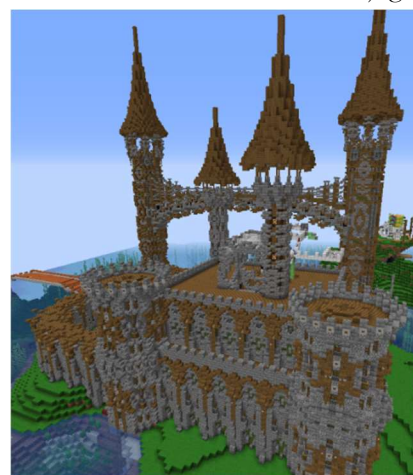


Fig. 6. Castell construït al Minecraft.
Elaboració pròpia

² Minecraft: videojoc en tres dimensions molt popular on el jugador té gran varietat de formes de jugar, com dissenyar i construir, sobreviure i viure aventures o *Player vs Player*. És un joc de món obert que ofereix moltes possibilitats per ser creatiu, semblant a un Lego virtual (vegeu Fig. 6).

Finalment, pel que fa als simuladors d'esports, en principi no tenen efectes psicològics remarcables, però s'estan contemplant com una manera d'ensenyar educació física als nens. Tanmateix, el jugador realitza un exercici de menys intensitat que si practiquéss l'esport que està simulant [10].

2.2 Efectes negatius

S'han investigat els efectes psicològics negatius de videojocs de tots els gèneres, però s'ha aprofundit sobretot en l'impacte dels videojocs violents a causa de la presència explícita d'agressivitat en aquests i que el 89% dels videojocs del mercat contenen elements de violència [11].

2.2.1 Violència

Diversos estudis realitzats en nens, adolescents i adults joves han conclòs que l'exposició a videojocs violents es correlaciona significativament amb una acceptació d'agressions físiques i empatia reduïda [12][13], el que s'anomena dessensibilització. Per exemple, en un estudi sobre els efectes a curt termini de videojocs violents amb relació als comportaments d'ajuda, 320 estudiants van ser assignats aleatòriament per jugar o bé a un videojoc violent o bé a un de no violent i, a continuació, van escoltar una lluita simulada on una persona acabava ferida. Els temps de les reaccions per ajudar van ser cronometrats i estipulats segons la probabilitat d'ajudar. Els resultats van indicar que els estudiants que havien jugat videojocs violents van trigar un 450% més que els altres alumnes per respondre al problema. Els autors van suggerir que també els mitjans de comunicació violents poden crear insensibilitat al dolor o patiment dels altres. Malgrat això, en aquest estudi no es van tenir en compte les variables personals addicionals, com podrien ser la voluntat d'ajudar o els nivells d'empatia previs al joc.

Hi ha molts estudis amb relació a la creença que els videojocs violents comporten comportaments violents. Per exemple, en un estudi van trobar alts nivells d'hostilitat i ràbia en adolescents que havien estat exposats llarga estona a jocs violents [12]. Molts estudis han trobat resultats semblants i en qualsevol edat, i conclogueren que jugar a jocs on la violència és la solució ensenya aquest mateix missatge. Els efectes a llarg termini en nens són causats pel seu aprenentatge basat en observació i imitació, així que els jugadors més joves pateixen més efectes a llarg termini. En canvi, en tenen menys a curt termini, ja que la seva xarxa de memòria associativa està subdesenvolupada. També cal assenyalar que tot i que els jugadors prefereixen videojocs realistes, els avanços tecnològics i un augment de realisme no incrementen els efectes negatius. Es va dur a terme un estudi per veure els efectes de recompensar accions violentes en videojocs amb variables relacionades amb l'agressivitat [14]. Els investigadors argumentaren que recompensar actes violents en els

videojocs augmenta l'agressivitat a través d'augmentar els pensaments violents. També van indicar que aquest augment d'agressivitat no és degut a la competitivitat.

Malgrat això, també hi ha nombrosos estudis que conclouen que els videojocs no afecten l'agressivitat o tenen un efecte mínim. Moltes vegades els nens en qui es troba un augment d'agressivitat en jugar videojocs han crescut en un ambient hostil i és difícil assenyalar si els videojocs són una de les causes de la violència. Els infants agressius de formacions acadèmiques baixes van resultar més interessats en videojocs violents, que al seu torn comporten més apreciació i ús d'aquests jocs [15]. Ferguson ha argumentat que hi ha prejudicis a l'hora de realitzar els estudis sobre els efectes dels videojocs violents [16]. Assenyalava que molts dels estudis que han declarat efectes significatius deguts a la violència dels videojocs han utilitzat metodologies poc acurades i que el disseny de la majoria dels estudis no permet una consideració plena dels efectes dels jocs violents en l'agressivitat en la vida real. Altres investigadors han declarat que molta de la recerca sobre aquest tema utilitza mesures d'analitzar l'agressió insòlites que poden augmentar les estimacions sobre els efectes.

2.2.2 Altres efectes negatius

En temes d'efectes desfavorables per la salut en jugar videojocs, diversos estudis han destacat una relació entre sessions de joc prolongades i dolor físic, convulsions epilèptiques i problemes a les articulacions, músculs i pell per culpa d'estar moltes hores amb poc moviment i restar hores de fer activitats físiques. També s'ha associat el fet de jugar a videojocs amb rendiment escolar reduït [17], perquè el temps dedicat a jugar a videojocs redueix el temps disponible perquè l'estudiant pugui estudiar.

Un estudi suggereix que jugar a videojocs es pot associar amb comportaments de risc entre els 18 i els 26 anys [18]. Els subjectes que jugaven videojocs van tenir nivells més alts d'utilització de drogues i relacions més pobres amb els amics i la família. Com que el videojoc recompensa la violència i les relacions poc sanes aquestes possiblement poden acabar formant part de la normalitat del jugador. Alhora, en jugar a videojocs es té menys temps per a les relacions humanes reals, fent que les parts socials del cervell estiguin menys desenvolupades. Aquest estudi també va trobar que, entre les dones, l'ús de videojocs està associat amb menys autoestima. Els autors també diuen que els videojocs poden ser un factor de risc pels adults joves, ja que és el moment en què es formen les identitats personals. Un estudi del 2007 va indicar una correlació entre la freqüència de jugar a videojocs de curses amb conducció competitiva i un augment d'accidents de cotxe [19]. Aquest mateix estudi també va trobar que els homes són significament més propens que les dones a prendre riscos en situacions de trànsit simulades en ordinador.

Finalment, diversos estudis demostren que l'addicció als videojocs és real i està classificada com a trastorn [20]. No obstant això, no s'ha de confondre passió amb addicció als videojocs, ja que aquesta segona

comporta un abús i dependència, que es correlaciona amb pitjor vida professional, acadèmica, social i de família. Realment no hi ha un nombre específic d'hores de joc a partir de les quals la passió es converteix en addicció. Cal tenir en compte les circumstàncies que envolten el temps dedicat a jugar. Per exemple, si en comptes de realitzar les obligacions de família o de la feina es juga a videojocs, l'individu es podria considerar addicte.

3. Introducció als llenguatges de programació i als *Game Engines*

En aquest apartat s'explica en què consisteixen els llenguatges de programació i els motors de joc o *game engines*. També s'ensenyen els escollits per desenvolupar White Caves i el motiu de la seva elecció.

3.1 Llenguatges de programació

Un llenguatge de programació és un llenguatge formal³ dissenyat per construir programes informàtics i per organitzar algorismes i processos lògics, que després seran duts a terme per un ordinador o sistema informàtic. Amb aquest mitjà, es pot controlar el comportament físic i lògic del programa i la comunicació amb l'usuari humà [22]. Els llenguatges de programació estan formats per un conjunt de símbols i regles sintàctiques i semàntiques que defineixen la seva estructura i el significat dels seus elements i expressions. La programació és el procés d'escriure, provar, depurar i compilar (si és necessari) un programa informàtic [23].

A l'hora d'escriure en un llenguatge de programació se segueix la següent successió de passos [24]:

1. Problema o necessitat a complir
2. Diagrama de flux o algoritme (en pseudocodi)
3. Codificació (traducció d'un algoritme en pseudocodi al llenguatge de programació)
4. Traducció del codi font al llenguatge màquina (procés automatitzat gràcies als compiladors)

A continuació s'ensenyà un exemple de pseudocodi per ordenar una llista d'elements (L)⁴:

```
procediment Ordenar (L)
  k ← 0
  repetit
    intercanvi ← fals
    k ← k + 1
    per i ← 1 fins n - k fer
      si Li > Li+1 llavors
        intercanviar (Li, Li+1)
        intercanvi ← vertader
      si no
        si no
    fins per
  fins que intercanvi = fals
fi del procediment
```

³ Llenguatge formal: en matemàtiques, lògica i ciències de la computació, els llenguatges formals són llenguatges on els seus símbols primitius (alfabet) i regles per unir aquests símbols (gramàtica formal) estan formalment especificats [21].

⁴ Pseudo codi en castellà extret de:

<https://sites.google.com/site/riicomalenguajedeprogramacion/galeria/pseudocodigo>

La gran majoria de programes funcionen de la següent manera: l'usuari introdueix uns valors (input), que són llegits pel programa. Aquest processa els valors d'acord amb el seu algoritme (metodologia per a resoldre un problema i determinat pel codi en qüestió), i després de fer certes operacions dona una resposta (output).

A continuació es pot veure un exemple de codi escrit en el llenguatge de programació Python, on s'introdueix el valor 2 (input), es multiplica per 5 (processament) i es dona 10 com a resultat (output).

```
1. In [1]:      input = 2
2.           result = input * 5
3.           result
4.
5. Out [1]:    10
```

Actualment hi ha milers de llenguatges de programació, però només entre deu i quinze es consideren populars, és a dir, que compleixin els següents requisits [25]:

- Hi hagi molta quantitat de *software* contemporani que utilitzi aquests llenguatges de programació.
- Siguin utilitzats per un gran nombre de persones.
- Hi hagi una comunitat activa i un mercat important per aquest llenguatge.

Actualment els més populars són els següents [25]:

- C
- C++
- C#
- Java
- JavaScript
- Perl
- PHP
- Python
- Objective-C
- Ruby
- Visual Basic

Aquesta llista canvia cada dècada aproximadament. No obstant, per què un programador hauria d'aprendre tants llenguatges si al cap i a la fi tots fan el mateix: escriure instruccions senzilles per ser executades dins d'un ordinador? No existeix un llenguatge únic per poder donar instruccions directament a l'ordinador?

Bé, realment aquest llenguatge sí que existeix, però no és cap dels mencionats anteriorment [25],[26]. Podríem dir que la CPU és el cervell d'un ordinador, és l'aparell que s'encarrega de processar totes les informacions. Tot i que podríem pensar que quan programem un programa el que fem és escriure codi perquè el processador pugui entendre la nostra voluntat, realment hi ha un pas intermedi. Nosaltres no escrivim manualment el codi que el processador entén perquè és pràcticament impossible. Són instruccions que treballen al nivell més bàsic de l'ordinador, així que són intel·ligibles per una màquina, però no per una persona [27]. A més, cada model de processador té el seu propi llenguatge. Així doncs, els llenguatges de programació com els que he enumerat anteriorment són un mitjà perquè la màquina i el programador s'entenguin, i els compiladors i intèrprets són les parts de l'ordinador que tradueixen entre els dos llenguatges [26]. El conjunt de números i lletres de la Figura 7 és un exemple de les instruccions que interpreta realment el processador del nostre ordinador [25].

```
00 00 00 01 40 00 00 00 00 00 00 01 00 00 00 00 5F 65 78 69 74 00 90 00
00 14 00 00 00 00 00 00 00 01 7A 52 00 01 78 10 40 11 40 5F 70 72 69 72
01 10 0C 07 08 90 01 00 00 2C 00 00 00 00 1C 00 6E 00 50 4E 58 41 72 67
42 52 00 00 11 40 64 79 6C 64 5F 73 74 75 62 5F 00 55 65 6E 76 69 72 6F
65 00 6C 02 00 00 00 03 00 A0 1D 00 03 00 E0 1D 68 65 06 1C 64 65 5F 6F
6E 00 50 4E 58 41 72 67 00 55 65 6E 76 69 72 6F 6E 00 67 00 02 6D 68 5F
```

Fig. 7. Exemple de codi màquina. Extret de: https://youtu.be/RKH5nSi_FuM

Alguns llenguatges de programació estan molt a prop del codi màquina (vegeu Fig. 8), i el més proper s'anomena *llenguatge ensamblador*. A partir d'aquest tenim llenguatges cada vegada de més alt nivell. En general, quan més a prop està un llenguatge del codi màquina, més difícil és d'escriure i més complicada és la seva sintaxi, i s'anomena *llenguatge de baix nivell*. Els llenguatges que estan allunyats del llenguatge ensamblador s'anomenen *llenguatges d'alt nivell*

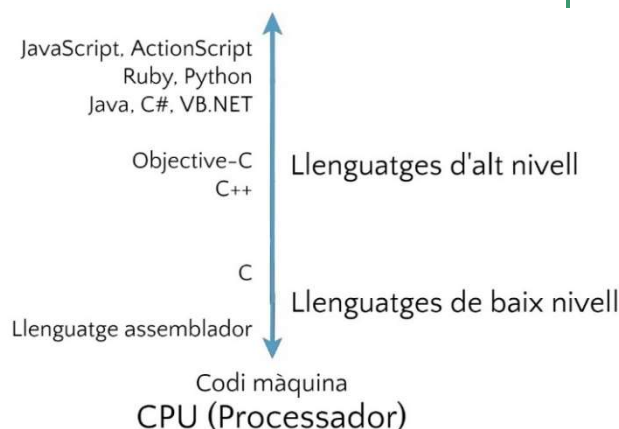


Fig. 8. Esquema dels llenguatges d'alt i baix nivell. Informació extreta de: https://youtu.be/RKH5nSi_FuM

[25]. El seu codi acostuma a ser més fàcil de compartir fins i tot entre diferents plataformes (és a dir, entre diferents equips informàtics amb diferents arquitectures de processadors) i acostumen a ser més fàcils d'escriure (ja que la seva sintaxi s'allunya del codi binari i s'escriuen amb ordres semblants a quan parlem, imitant el llenguatge i la comunicació humana). No obstant, els llenguatges d'alt nivell normalment són més lents d'executar, ja que han de fer una traducció més complexa fins a arribar al llenguatge de la CPU, però avui en dia aquestes diferències són mínimes en la majoria dels casos, així que normalment es recomana aprendre un llenguatge d'alt nivell.

El llenguatge de programació utilitzat en el videojoc d'aquest treball és C# (llegit *ci-sharp*), un llenguatge de programació desenvolupat i estandarditzat per l'empresa Microsoft com a part de la seva plataforma .NET [28]. C# és un dels llenguatges de programació dissenyats per la infraestructura del llenguatge comú (una especificació estandarditzada que permet que aplicacions escrites en diferents llenguatges d'alt nivell es puguin executar en diferents plataformes tant de hardware com de software sense que s'hagi de reescriure ni recompilar el seu codi font). La seva sintaxi bàsica deriva del llenguatge C++, que al seu torn prové de C, i utilitza el model d'objectes de la plataforma .NET, que és similar al de Java, però inclou millores provinents d'altres llenguatges. La seva programació és orientada a objectes (POO). Aquest tipus de programació permet al programador crear objectes informàtics que tenen propietats i accions [28]. És a dir, es pot crear un objecte i fàcilment assignar-li valors que podrien correspondre a la seva vida, la seva puntuació, la seva experiència, etc.

C# és un llenguatge molt popular així que és fàcil trobar-ne cursos i fòrums per preguntar dubtes. A més a més, aquest és el llenguatge per defecte en Unity, tal com s'explica en el següent apartat. Per tant, C# és un dels llenguatges més usats en videojocs i el que s'ha utilitzat per programar White Caves.

3.2 Motors de Joc

Els motors de joc (o *Game Engines* en anglès) són aplicacions de programari que permeten dissenyar, crear i representar un videojoc eficientment. Abans d'haver-hi motors de joc cada joc es feia des de zero, així que els desenvolupadors dedicaven molt de temps a fer feines que ara ens podem estalviar gràcies a aquests programaris. En la majoria de casos els motors de joc ofereixen un conjunt d'eines de desenvolupament visuals i components de programari reutilitzables [29],[30].

Els motors de joc s'encarreguen de fer més fàcil i eficaç la majoria de passos involucrats en el desenvolupament d'un videojoc, i en el cas que es facin en programes externs facilita la unificació dels següents aspectes per tenir un bon producte final.

3.2.1 Gràfics

Els motors de joc actuals poden suportar gràfics impressionants perquè proporcionen al desenvolupador una arquitectura amb gran capacitat per renderitzar i accés a *fast graphics API*⁵ (vegeu Fig. 9 com a exemple) [31].

Normalment també tenen el seu propi programari per a desenvolupar *assets*⁶ o importar-los des de programes externs.

3.2.2 Àudio

Els desenvolupadors poden dissenyar els seus sons des de zero, començant amb la composició musical, efectes de so, actuació de veu i acabant amb la

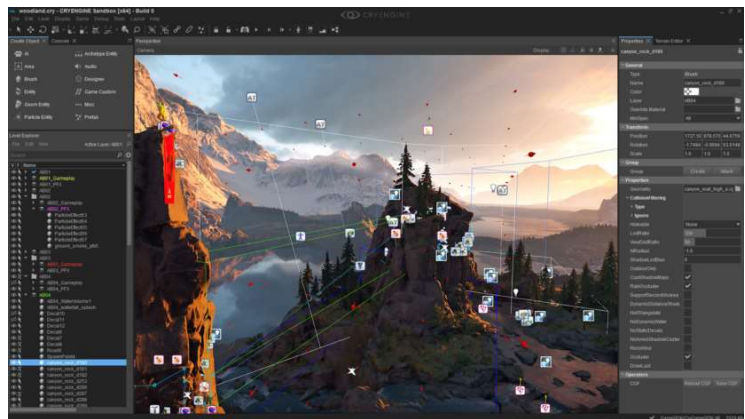


Fig. 9. Exemple de renderització de gràfics en CRYENGINE [Captura de pantalla].

Extret de: <https://www.geckoandfly.com/32489/game-engines/>

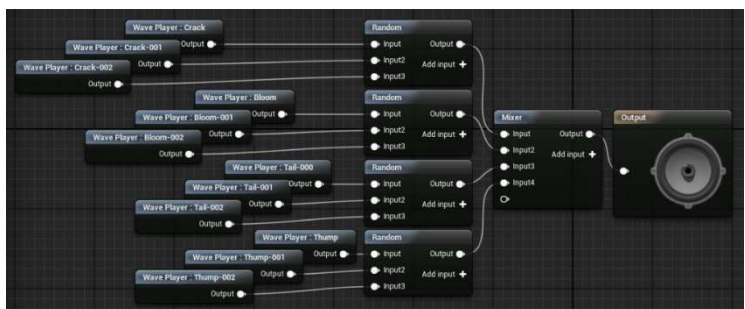


Fig. 10. **Procedural explosion.** Exemple de software intern a un motor de joc per editar i crear àudio.

Extret de: <https://www.asoundeffect.com/procedural-game-sound-design/>

⁵ API: De l'anglès, *Application Programming Interface*, és una eina utilitzada per programadors per ajudar-los a crear programari. Permeten al programador lliurar solucions bastant ràpidament.

⁶ Asset: Contingut utilitzat en la creació de videojocs o altres programes d'ordinadors, com per exemple textures, models 3D, efectes de so i música, text, diàlegs o programari.

postproducció (vegeu Fig. 10 com a exemple) [31]. També poden comprar *assets* de so des de plataformes de compra i venda de programaris i *assets*, com la Unity Asset Store⁷.

3.2.3 Creació de xarxes

A molts jugadors els agrada jugar amb amics o fins i tot desconeguts d'Internet, però per certes classes de jocs la creació de xarxes pot necessitar molta potència del servidor i això pot resultar en processos molt complicats [31]. Per això molts motors de joc també ofereixen la possibilitat de crear xarxes per jugar *online* molt més fàcilment.

3.2.4 Físiques

El sistema de físiques en els motors de jocs ofereix components que gestionen les simulacions físiques. D'aquesta manera no s'ha de programar cada moviment de cada objecte per separat i ni tan sols reprogramar cada vegada les línies de codi que simulen la gravetat o la col·lisió d'objectes, sinó que permeten crear objectes que es comporten de forma realista en cap o molt poques línies de codi [31].

3.2.5 *Graphical user interfaces (GUI)*

La majoria de motors de jocs permeten als desenvolupadors crear la seva pròpia GUI o Interfície Gràfica d'Usuari per tal que lligui amb l'estil del joc, el seu art i la seva història (vegeu Fig. 11 com a exemple). Ofereixen eines com botons o menús desplegable, diapositives i formes de combinar diferents elements amb els quals es pot interactuar.

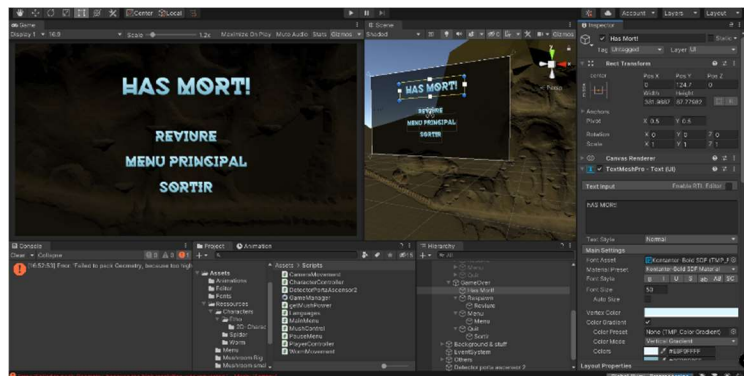
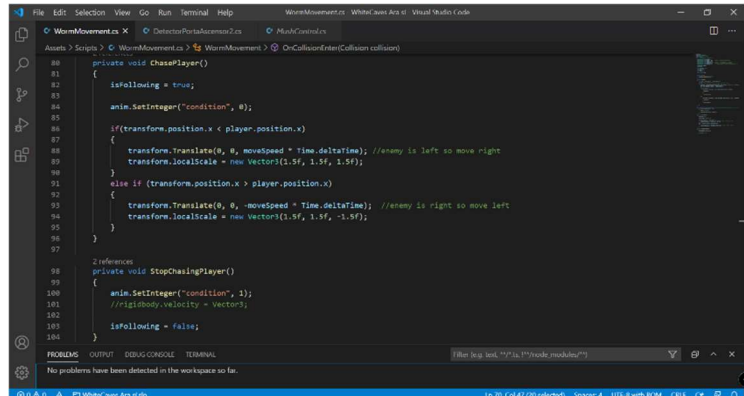


Fig. 11. *Graphical User Interface* de Unity. Exemple de software intern a un motor de joc per editar i crear pantalles, botons, desplegables, etc. Elaboració pròpia.

⁷ Unity Asset Store: exemple de botiga de compra i venda d'assets: <https://assetstore.unity.com/>

3.2.6 Programació

Un aspecte molt important dels motors de joc és la programació o *Scripting*⁸, que es pot utilitzar per definir la lògica dels components del joc afegint comportaments. Alguns motors de joc ofereixen *scripts*⁹ preprogramats per tal que sigui fàcil crear els controls de la càmera, la relació entre els elements del joc i el sistema d'animacions que es pot utilitzar per controlar l'animació dels personatges. També poden incloure llibreries de codi per a fer més directe la programació de certes accions (vegeu Fig. 12 com a exemple de programació utilitzant la llibreria¹⁰ Unity.Engine, creada pel motor de joc Unity).



```
80 private void ChasePlayer()
81 {
82     isFollowing = true;
83
84     anim.SetInteger("condition", 0);
85
86     if(transform.position.x < player.position.x)
87     {
88         transform.Translate(0, 0, moveSpeed * Time.deltaTime); //enemy is left so move right
89         transform.localScale = new Vector3(1.5f, 1.5f, 1.5f);
90     }
91     else if (transform.position.x > player.position.x)
92     {
93         transform.Translate(0, 0, -moveSpeed * Time.deltaTime); //enemy is right so move left
94         transform.localScale = new Vector3(1.5f, 1.5f, -1.5f);
95     }
96 }
97
98 //reference
99 private void StopChasingPlayer()
100 {
101     anim.SetInteger("condition", 1);
102     //isObsoy.velocity = vectors;
103     isFollowing = false;
104 }
```

Fig. 12. Programació del moviment del cuc de White Caves, a Visual Studio Code i utilitzant la llibreria de Unity.Engine. Elaboració pròpia.

Hi ha molts programes que es podrien utilitzar com a motors de joc, però per aquest treball s'ha utilitzat Unity 3D perquè és gratuït i amb una comunitat d'usuaris molt activa, és a dir, és fàcil trobar manuals i fòrums per aprendre i solucionar dubtes. A més a més, és dels programes més complets, i alhora té un aspecte bastant intuïtiu, que permet ser utilitzat tant per professionals com per aprenents.

⁸ Scripting: un altre terme per referir-se a la programació; acte d'escriure codi [32].

4. Realització d'un videojoc

En aquest apartat s'explica resumidament la teoria sobre la creació de videojocs i, amb més detall, la meua experiència en crear *White Caves*. L'alternança entre part teòrica i part pràctica està feta a propòsit. Mancaria de sentit fer una part teòrica molt extensa sobre el disseny i desenvolupament d'un videojoc, perquè a internet es poden trobar bones explicacions de professionals d'aquest camp. Així doncs, l'apartat resumeix la teoria i és un diari de la meua experiència. Està estructurat en dues parts: el procés de disseny i el de desenvolupament o producció. Tot i que no és necessari, s'aconsella haver jugat prèviament a *White Caves*¹¹ per comprendre millor les explicacions i descripcions. En qualsevol cas, per ajudar a entendre-les, es poden trobar captures de pantalla del joc a l'Annex 2.

4.1 Disseny

Dissenyar el guió d'un videojoc, juntament amb el seu apartat artístic, les seves mecàniques i els seus personatges, no és tan fàcil com seguir una sèrie de passos. Si no es planteja bé el disseny des d'un inici (tal com va passar amb la primera història que vaig crear, explicat a continuació) apareixen problemes en el desenvolupament del joc. Per crear *White Caves* s'ha seguit el plantejament més utilitzat (descriu en aquest apartat) per dissenyar-lo eficientment i evitar problemes a la llarga. Consisteix en començar la història i disseny a grans trets i, poc a poc, anar especificant més i més, sempre tenint en compte la interacció i la dinàmica jugador-personatge, explicades més endavant.

4.1.1 La història

Per aprendre a dissenyar la història s'ha utilitzat principalment un blog en línia [33] i, puntualment, altres recursos com revistes de videojocs, altres pàgines d'Internet, petits documentals o debats informals amb amics coneixedors del tema. Aquests últims recursos, tot i que han ajudat a tenir una idea més clara del que comporta el disseny de la història i el joc sencer, s'han posat en pràctica només parcialment. Inicialment es va fer l'error de no seguir un plantejament pautat i es va obtenir una història de mala qualitat, com s'argumenta més endavant. Per crear la nova història es van seguir els passos descrits en aquest apartat.

¹¹ *White Caves* es pot descarregar gratuïtament des de bit.ly/WhiteCaves

4.1.1.1 *La clam d'una bona història*

Cada videojoc és diferent, i tracta de manera diferent la història. Hi ha videojocs que duren més de 100 hores però que el jugador no té opcions per triar i, en canvi, n'hi ha que duren 20 minuts i el jugador té moltes eleccions a fer. A més a més, cada jugador té preferències diferents amb relació al pes de la història.

Tot i això, hi ha un aspecte que molts videojocs tenen en compte, la dinàmica jugador-personatge. Aquesta norma diu que el jugador i el personatge principal són un, així que no s'ha de fer dur a terme al personatge principal una acció que el jugador no faria. El jugador vol sentir que té control, cosa que el diferencia de l'espectador, més passiu, d'una pel·lícula. Per tant, el jugador ha de poder interactuar amb l'entorn, tenir acció directa i veure la repercussió de les seves accions, fins i tot quan sembla que no hi ha eleccions per fer ni una història clara.

El dissenyador Sid Meier va definir el joc com “una sèrie de decisions interessants” [34]. És cert, el disseny de videojocs es caracteritza per crear problemes plaents sobre la presa de decisions, segons qüestions estratègiques, dilemes morals o bifurcacions narratives. En els videojocs s'acostuma a emfatitzar el poder de decisió com un plaer essencial del jugador: “ets lliure, tu decideixes, tot depèn de tu”. S'ha de tenir en compte, però, que aquestes utopies de poder no són políticament neutrals, i és que hi és molt present el discurs neoliberal i el mite del lliure albir, que la ciència moderna desmenteix. Això no obstant, també hi ha casos de videojocs *indies* que sabotegen aquest plaer de decidir, com per exemple, *Papers, Please*, que tracta sobre la impotència d'un inspector d'immigració en un país en conflicte [34].

Inicialment, en el meu primer intent de crear un videojoc, es va dissenyar una història que no complia el requisit de tenir una dinàmica definida jugador-personatge i, per tant, mancava *jugabilitat*. Aquest és un resum de la història d'aquest prototip de videojoc, situat temporalment a l'edat mitjana:

El personatge principal desperta, sense records previs, en una habitació fosca, però gràcies a la llum que entra per la finestra troba un diari sobre la tauleta de nit. En llegir-lo descobreix que és el seu i, doncs, qui és.

En sortir de l'habitació es troba gent morta pels passadissos de l'edificació, que resulta ser una taverna. També, tirat per terra, troba el pin d'un escut d'una família aristòcrata.

Després de no haver trobat ningú viu surt de la taverna. És de nit, amb lluna plena i alguns núvols, però s'aconsegueixen veure una mica les estrelles. Segueix el camí que passa per davant l'edifici fins a trobar un carro de fusta, sense cavalls i amb persones mortes dins. També troba un binocle que resulta ser un ítem que, en utilitzar-lo per veure el cel, les estrelles s'uneixen amb línies formant constel·lacions, que representen els camins per on pot caminar el jugador. Així doncs, les constel·lacions del cel fan d'equivalent a un mapa, i els núvols amaguen les zones del món encara no descobertes.

El personatge segueix caminant pel camí fins a arribar a un castell, on veu un tapís penjat del mur amb el mateix escut dibuixat que havia trobat a la taverna. Escriu noves notes al diari: suposa que els assassins de totes aquestes persones han sigut els homes del castell, i que un dels pins de la seva roba deu haver caigut a terra. Rodeja el castell i troba una petita escaleta per on pot entrar.

En accedir a l'interior es troba una petita taula amb un calaix i, a dins, unes claus. Si el jugador vol les pot agafar. Segueix caminant i es troba una porta, corresponent a l'oficina principal del castell. En el cas d'haver recollit la clau anteriorment pot accedir a una petita biblioteca on el jugador trobaria eastereggs¹² sobre la creació del videojoc.

El jugador surt de l'estança i camina pel passadís principal fins a trobar un pati interior. A la cantonada, a les fosques, veu un soldat escrivint en un llibre, però sent que algú arriba i se suïcida allà mateix. El jugador s'apropa i llegeix el llibre. Es tracta del diari d'aquest home, soldat de l'aristòcrata que s'encarrega del castell. Explica com ella estava farta que la tractessin com una nena i volia demostrar que podia governar perfectament sense el seu pare o germà, morts a la guerra. Com que no aconseguia que la respectessin va fer matar la gent del poble per aconseguir respecte a través de la por. Morta la gent del poble va fer matar també alguns dels de la cort. El castell es va quedar molt sol, i per això el jugador havia aconseguit entrar al castell sense ser vist.

Al soldat, naturalment, l'havia afectat molt això i, juntament amb efectes de les drogues que els havien posat al menjar perquè complïssin el manat, s'havia espantat en sentir la presència d'algú i s'havia suïcidat perquè no el torturessin.

El jugador ha d'abandonar el castell el més ràpid possible, però no pot baixar per on ha entrat. Un altre soldat el veu i el tenca a la presó, on s'adorm. Temps després es desperta amb el soroll d'una porta obrint-se. La cel·la té la mateixa estructura però és moderna, i la persona que apareix per la porta va vestida com en l'actualitat i no com a l'edat mitjana. S'asseu al seu costat i li explica que ha format part d'un experiment per veure com és d'immersiva una nova tecnologia de realitat virtual que estan dissenyant; res del que acaba d'experimentar és real.

La pantalla es torna negra i s'acaba el joc.

Aquest guió no és bo per a un videojoc per a diverses raons:

- Hi ha preguntes que no tenen una bona resposta, però que són claus perquè l'estructura de la història es mantingui. D'altra forma, pot ser molt confusa pels jugadors. El protagonista té amnèsia? Qui és i què fa allà? Per què? Està bé que sigui anònim, però s'ha de donar una mica d'informació sobre ell (no n'hi ha prou amb el diari) perquè el seu paper tingui coherència i pugui alterar els esdeveniments.

¹² Easteregg: en els videojocs, es coneix el terme *easteregg* com a objecte, missatge, característica o sorpresa amagada al jugador i que no és necessària per a passar-se el joc [35]. Normalment tenen a veure amb els desenvolupadors i no tant amb la història del joc en sí.

Per què el protagonista ha sigut l'únic que no han matat? Han assassinat als altres de la taverna, ell també hauria d'estar mort. Com pot ser que tots els esdeveniments hagin passat a la mateixa nit? Han matat tant la gent del poble com la de la cort en una sola nit? I el protagonista sense assabentar-se de res? I encara queda temps perquè el jugador es desperti i succeeixi tot el joc? Com ha aconseguit la noia noble matar a tots els seus ciutadans i fins i tot membres de la cort sense cap revolta efectiva? No hi havia millors formes de guanyar-se el respecte? És clar que el fil històric trontolla.

- En el moment en que el jugador es troba la clau, realment no té dues opcions. Es pot dir que té l'opció d'utilitzar-la o no, però això realment no és una opció, ja que no afecta en res al joc. Agafar-la és la reacció natural, com també utilitzar-la. Una elecció real seria si hi hagués dues portes, per exemple. Ambdues podrien arribar al mateix lloc, però l'experiència seria diferent. Una podria ser més llarga i l'altra més difícil.
- Els *eastereggs* agraden molt, però realment no és una bona idea posar-los tant al principi i a l'abast. El jugador ha de ser absorbit pel joc, i trobar-se un element extralingüístic poc després de la introducció no ajuda a la immersió del jugador, ja que és anticlimàtic i pot arruïnar l'experiència, i és una història que pretén ser seriosa. Es podria fer més complicat l'accés a aquesta sala dels *eastereggs* o posar-los de forma molt indirecta (és a dir, que siguin fàcilment perceptibles però no el seu significat).
- Una de les pors que es tenia a l'hora de dissenyar aquesta història era fer-la massa realista i que després s'arruïnés amb alguna absurditat. I és cert que s'ha de vigilar amb aquest tema, però tampoc en extrem perquè si no quedaria una història amb molt poc encant. Per exemple, Hideo Kojima, el creador de Metal Gear i Death Stranding, dos jocs molt populars, fa reflexions dins dels jocs sobre temes tan complexos com el benefici comú dels conflictes bèl·lics. No obstant, en una ocasió acaba apareixent un monstre que és un home calb obès amb patins, que, efectivament, xoca al jugador, però no suficient com per arruïnar el joc, que no perd la seriositat.
- Hi ha molt de text. S'havia intercanviat la propietat interactiva dels videojocs per molta lectura. No es tracta d'un llibre ni d'una pel·lícula.

Era clar, doncs, que aquesta història era molt feble, així que era necessari crear-ne una altra. En aquesta ocasió, tal com s'hauria d'haver fet des d'un començament, es van seguir els passos que es descriuen en aquest apartat de disseny.

4.1.1.2 Definir el fil històric a grans trets

Després d'aquest intent de videojoc es va tornar a començar de zero amb el disseny, seguint els següents passos, extrets del mateix blog en línia mencionat anteriorment [33].

1r: tenir clar quina és la meta o significat d'aquest videojoc. Per què es vol fer, què es vol que el jugador experimenti, el sentit de la història, què es vol reivindicar o quin és el missatge que es vol que el jugador capti.

2n: pensar les “normes” del món, les quals són seguides pels personatges, els objectes, i fins i tot el jugador quan les entén. Totes les accions importants han de mantenir certa coherència. No obstant, trencar les normes en moments puntuals és un recurs difícil d'utilitzar però que de vegades aporta qualitat a la història. Aquestes no són simplement normes legals del món o lleis de la física, també ho són el comportament dels personatges i el fil històric (per exemple mancaria de sentit, si no s'explica, que un videojoc comenci amb un *shooter* a una estació espacial i de sobte canviï la temàtica i l'ambientació i passi a ser un joc de rol a una casa de Londres).

3r: quan ja es tenen bastant clares les normes (no necessàriament s'han de tenir completament definides) s'ha de pensar en la història en si. Si les normes es pensessin una vegada la història ja s'ha creat, de vegades per tal que la història es mantingui, s'acabaria amb incoherències o normes il·lògiques i incomprensibles, com és, en part, el cas del primer prototip de videojoc que he fet, mencionat anteriorment. El pas més important és escriure el guió creant el contorn del fil històric general. És a dir, escriure els successos generals de cada personatge principal, independentment de les seves decisions. A l'hora de definir els detalls pot resultar difícil, però en fer els següents passos serà més fàcil precisar.

En aquest cas, es comparteix l'opinió de Sand Bloom Studio quan diuen que un videojoc és una forma d'expressió artística i, doncs, també és una bona eina de crítica social on s'hi pot incloure un profund missatge, que no dicta al jugador el que ha de pensar, sinó que li fa qüestionar certs aspectes.¹³ Així doncs, a l'hora de pensar la història, els personatges i el món, s'ha de tenir en compte que fins i tot els colors d'una samarreta poden simbolitzar emocions.

- El missatge que es vol transmetre en aquest videojoc suggereix que els problemes personals, tant mentals com físics, tenen solució, i la que es proposa és de mirar-se a si mateix, analitzar-se i treballar a millorar-se, no pel que pensaran els altres, sinó per la pròpia satisfacció. Com a tècnica d'anàlisi i millora personal se suggereix el mètode Wim Hof, una combinació de respiració, exposició al fred i meditació, que rep el nom del seu autor [36].

Com a missatges secundaris es proposa que per enfrontar un problema no fa falta enfrontar-lo violentament i directament, sinó que es pot solucionar, per exemple, canviant hàbits; que certa

¹³ Entrevista a Sand Bloom Studio inclosa a l'annex.

persona o idea que en un principi s'havia menyspreat i rebutjat des de la desconexença pot resultar ser una sorpresa positiva; que la serenitat i l'amor acompanyen en aquest viatge de descobriment personal (simbolitzades pel taronja en la vestimenta del personatge principal).

Malauradament, aquests missatges no són perceptibles en la versió prematura del joc que es va fer pública el 2 de novembre, que per falta de temps no va poder incloure tot el que es tenia dissenyat. En la versió completa, es fa ús de diverses metàfores per referir-se a aquest procés de superació personal. Per exemple, l'eufòria necessària per dur a terme aquest trajecte és representada per la joventut del personatge (independentment de l'edat real de la persona que es troba darrere la pantalla), i el viatge metafòric per solucionar el problema en qüestió és representat per les coves. Les coves tendeixen a seguir un recorregut lleugerament ascendent, que es fa totalment vertical quan s'arriba a l'etapa final (el temple de les profunditats). Aquesta ascensió simbolitza la fugida dels mals hàbits i problemes, la part fosca de les coves, per arribar a la superfície, blanca de neu i boira, que representa la llibertat i el benestar.

Certes zones del joc són fàcils, però altres són molt frustrants (tal com han corroborat els jugadors de la *demo*¹⁴). La frustració experimentada pels jugadors és intencional. La superació personal no és un tema fàcil, pot ser molt devastant, i es vol que el jugador experimenti part d'aquest sentiment. Per fer més forta aquesta associació del personatge amb una persona real i imperfecte (amb oposició a la majoria de videojocs on el protagonista és un heroi superhumà) s'ha utilitzat sales molt més grans que el personatge (el protagonista apareix, doncs, com un petit conjunt de píxels en una enorme sala que ha de superar). En *Shadow of the Colossus*, per donar aquest mateix sentiment, quan un gegant trepitja a prop del protagonista, aquest cau a terra, resta allí alguns segons i posteriorment s'aixeca (deixa clar que el gegant és poderós, però el protagonista és dèbil i haurà de lluitar molt fort per aconseguir el seu objectiu).

- Les normes de *White Caves* són, doncs, que si es presenta un problema, com un enemic, no se l'ha de matar lluitant amb una espasa, però tampoc se n'ha de fugir. S'ha d'afrontar amb enginy resolent puzzles. Així doncs, no hi ha cap mecànica de lluita.

No és un joc realista. El jugador salta molta altura i no es fa mal en caure, hi ha un cuc gegant, plataformes de pedra mantingudes suspeses en l'aire i bolets gegants. De la mateixa manera qüestions com "Com ha arribat el protagonista a aquesta cova?", "Les mines abandonades no tenen un accés realista" o el fet que cada plataforma estigui perfectament posada perquè es pugui realitzar

¹⁴ (Versió) demo: prototip, versió incompleta i d'avaluació d'un programa informàtic per tal de mostrar el funcionament i les seves funcions.

aquell salt no distreuen el jugador ni li trenquen l'experiència de joc, ja que en cap moment pretén ser un joc realista, encara que tracti un tema molt seriós.

- I per últim, el meu fil històric es podria resumir així:

Un noi jove es troba en les profunditats d'una cova ben fosca. A mesura que el jugador avança es va trobant petits reptes i recompenses. Es troba un cuc gegant, del qual n'ha d'escapar. Una aranya, possible enemic malinterpretat, es converteix en la seva amiga, que l'ajuda a sortir d'aquell infern. En arribar a un temple que es troba a les profunditats de la terra, aconseguixen ascendir ràpidament fins a la superfície, on tot està cobert d'una blanca i freda neu, que acaba congelant l'acompanyant del personatge.

4.1.2 Definir el món i els ítems

El món és essencial en la majoria de videojocs; situa el personatge cronològicament i físicament, estableix les normes, etc. En aquest cas, s'ha definit el món al mateix temps que es realitzava el següent apartat, que correspon al disseny dels personatges, ja que són dos temes que van molt lligats. Per aprendre a dissenyar correctament el món, s'ha fonamentat en part en el mateix blog en línia que anteriorment [37], però diversos passos no s'han aplicat, ja que no s'escau per aquest estil de videojoc.

En primer lloc, s'ha decidit quin és l'abast o llibertat dels jugadors per moure's pel mapa:

- Si és un joc de món obert, el jugador té pràcticament plena llibertat per anar d'un lloc a l'altre, i tot i que la qualitat dels detalls varia en cada títol, els jocs de món obert prioritzen l'exploració. Grand Theft Auto V, per exemple, té un mapa molt gran i posa molt detall en les façanes dels edificis, però en canvi no et permet entrar en la majoria d'ells. Tot i així, sí que hi ha jocs que et permeten explorar tot el terreny i tenen en compte fins i tot el més petit detall, com Skyrim o Breath of the Wild.
- En canvi, si és un joc de món lineal, el jugador no té tantes opcions per anar d'una posició a una altra, sinó que hi ha un camí específic (o més d'un). L'exploració segueix sent possible en molt d'ells, com en *Bioshock*, on hi ha moltes àrees per explorar però no és permès navegar per qualsevol lloc ni retornar d'on es prové. White Caves és clarament lineal. Fer un joc de món obert i que sigui entretingut en la majoria de casos requereix molt de temps i experiència que no es tenia. A més a més, per aquest fil històric s'esqueia més un món lineal. Hi ha certs moments en què el jugador pot anar per dos camins, però ambdós es retroben poc després.

El món de White Caves és en tres dimensions espacials, decisió que es tenia clara des d'un inici, ja que un dels motius de voler fer un videojoc era la meua passió per crear models 3D. No obstant, es tracta d'una perspectiva poc convencional: els models dels personatges, món i objectes són models en 3D, però la càmera només es mou en dues dimensions. Com a referència es va utilitzar el videojoc anomenat "Inside" i desenvolupat i publicat per Playdead el 2016 (vegeu figura 14). Té una estètica senzilla però, alhora, produeix un sentiment d'horror i pena al jugador pel protagonista. El següent diagrama compara els tres tipus d'utilització més comuns de les dimensions espacials en els videojocs. No hi són presents els videojocs amb quatre o més dimensions espacials, ja que són extremadament difícils de comprendre i n'hi ha pocs exemples.



Fig. 13. **Gris**, un joc en 2D que situa els diferents sprites davant o darrera per donar sensació de perspectiva. Fotografia del joc.

Extret de: <https://www.pcgamer.com/au/gris-review/>



Fig. 14. **Inside**, un videojoc amb models 3D però que la càmera es mou únicament en dues dimensions. Fotografia del joc.

Extret de: <https://www.hobbyconsolas.com/guias-trucos/inside/como-encontrar-todos-orbes-ocultos-inside-acceder-final-secreto-180782>

Fig. 15. **Grand Theft Auto V**, un joc de món obert i totalment en 3D. Fotografia del joc

Extret de: <https://www.usgamer.net/articles/grand-theft-auto-v-ps4-review>



Una vegada es té clara l'amplitud del moviment i del món, s'ha de dissenyar un **rerefons històric**, que ajudarà a donar significat a objectes, personatges, interaccions, cultures, etc. És important per submergir-se en la seva història i món, d'altra banda, si no es té en compte el rerefons històric, semblarà que els objectes, ambient i personatges són buits de significat profund.

De la mateixa forma, per jocs extensos, es recomana dissenyar una o diverses cultures que habitin el món, establir una jerarquia, com es relacionen entre elles i quines diferències i semblances tenen. Això també donarà més vida i sensació d'un món complet. Molt sovint es miralla el conflicte entre diverses cultures, races o espècies en el videojoc amb alguns dels problemes socials del nostre propi món.

Totes les coves de White Caves representen sentiments i l'interior emocional del personatge. En aquestes coves hi ha restes de mines abandonades i d'un temple. També habiten certs animals, com unes aranyes. No obstant, White Caves té un rerefons històric molt breu, ja que el temps de joc és curt i gaire detall podria arribar a ser confús pel jugador, ja que no tindria temps de comprendre'l correctament.

També s'ha de crear els ítems com armes, eines, llibres, mapes, etc. Aquests són molt importants en molts jocs, ja que és una de les formes més importants d'interacció entre el jugador i el món.

En aquest cas, s'ha començat descrivint el seu ús. S'ha de tenir en compte on seran trobats (els hi donarà un NPC, estarà en un bagul o bé serà aconseguit directament pel jugador a partir de materials de l'entorn). És molt pràctic apuntar en un full de càlcul tota la informació sobre cada ítem, així com es podrà fer sobre els personatges i edificis. En el cas de White Caves, només hi ha un ítem, els bolets gegants:

La seva creació ha estat determinada principalment pel seu ús: era necessària una nova mecànica¹⁵ per donar més varietat al joc. Com que està centrat en saltar, s'ha acabat pensant en uns bolets gegants on s'hi pot saltar a sobre i, després, es pot arribar a una plataforma allunyada que sense el bolet hauria estat massa distant com per arribar-hi (vegeu Fig. 16 i 17). També serveixen com a atractiu per l'aranya, un dels personatges.

Representa que a l'aranya li agraden molt aquests bolets, i ella anirà allà on el jugador els tiri. D'aquesta forma es poden fer puzles bastant divertits i entretinguts.



Fig. 16. Model 3D del bolet de *White Caves*. Elaboració pròpia.

¹⁵ Mecànica (de joc): patrons d'interacció; regles que guien els moviments o accions del jugador i la resposta del joc a aquestes.

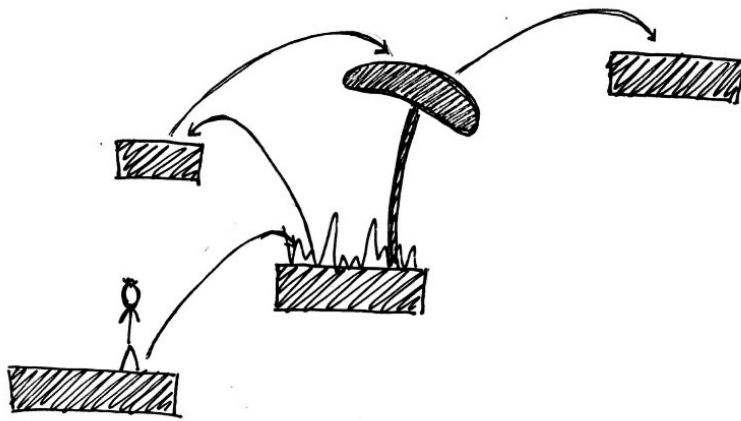


Fig. 17. Esquema de la utilitat dels bolets. Elaboració pròpia.

Els bolets es troben en una zona ombrívola de la cova, on n'hi ha molts de petits. Al passar-hi, s'obté el poder de plantar els bolets, però només en unes zones específiques: les plataformes que tenen herba, que indiquen que són fèrtils i el bolet hi podrà créixer (vegeu Fig. 18).



Fig. 18. Zona on s'aconsegueix el poder dels bolets. Elaboració pròpia.

Finalment, s'ha dibuixat el contorn del món. Primer, imaginant-lo, després en un paper amb poc detall i, a continuació en un paper amb més detall (vegeu figura 19. Les altres parts del mapa es troben a l'Annex 2). A partir d'aquest mapa del món, ha estat més fàcil el disseny dels ítems i el petit rerefons històric que té.



Fig. 19. Primera part del mapa provisional a paper de White Caves. Elaboració pròpia.

A continuació, s'ha dibuixat el mapa en un *canvas* digital infinit (vegeu figura 20. Les altres parts del mapa es troben a l'Annex 2). S'han canviat moltes coses respecte del disseny a paper, com el recorregut del jugador, els puzles o les posicions dels enemics. Es tingueren molts problemes amb l'escala i les proporcions, però al ser digital es va poder modificar fàcilment si es volia rectificar quelcom.

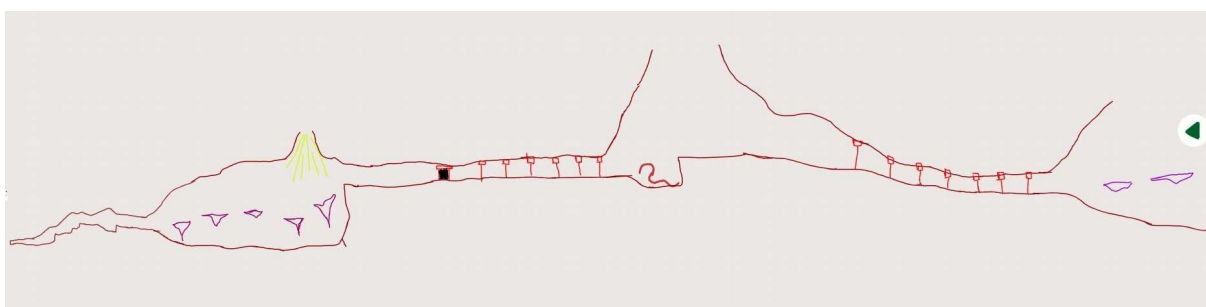


Fig. 20. Primera part del mapa provisional digital de White Caves. Elaboració pròpia.

4.1.3 Dissenyar els personatges

Una vegada s'ha tingut el món bastant clar, s'ha començat el procés del disseny de personatges. Per aprendre'n, s'ha utilitzat el blog escrit per E. M. Welsh "How to Write Video Game Characters" [38]. També es començà un curs de la pàgina de cursos en línia Coursera anomenat "Story and Narrative Development for Video Games" [39], impartit per l'Institut d'Arts de Califòrnia i com a professor en Dariush Derakhshani. No obstant, es va decidir que seria millor no acabar el curs, ja que impartia coneixements massa bàsics i s'ha pogut invertir aquest temps en el desenvolupament del videojoc.

Els personatges, sobretot els principals, són una de les formes de transmissió de missatges entre el videojoc i el jugador i, normalment, es valora que siguin realistes i que no deixin anar textos (una bona animació del model del personatge ajuda al fet que sigui més creïble, però és important que el personatge en si tingui un petit rerefons y una voluntat definida, igualment que després el jugador no l'acabi coneixent mai).

De vegades, el personatge principal està dissenyat en part pel jugador, però generalment no influeix en com és el personatge, com en el cas de White Caves. En el primer cas, el jugador té més facilitat en sentir-se part del joc, però un personatge ja definit, si està ben fet, no té per què complicar la immersió.

A continuació, s'ha fet una llista amb tots els personatges que es vol que apareguin. Naturalment, els personatges principals han tingut una descripció molt més extensa que els secundaris. S'ha donat un perquè al personatge -perquè està a la història, què aporta. També s'ha definit quin és el seu rerefons històric, què fa aquest personatge, en què influència a la història, la seva personalitat i la seva estètica. Respondre les següents preguntes ha ajudat a definir-lo [39]:

1. Qui és aquest personatge? És principal o secundari? Com influeix en la història?
2. Quin és el seu aspecte? Què porta a sobre?
3. Quina és la seva personalitat i com influeix en la història?
4. Com es relaciona amb altres personatges?
5. Quins són els seus punts forts i febles? Com afecten la història?
6. Quines són les motivacions i les voluntats del personatge?

A continuació, s'utilitza d'exemple les informacions de dos dels personatges principals de White Caves:

1. És Etho, el personatge principal, controlat pel jugador. Tota la història gira al seu torn, ja que és la història de la seva evolució personal.
2. És menut, aspecte amplificat per les grans coves que l'envolten, sobretot al principi. És optimista, aventurer i independent, caràcters simbolitzats pel seu cabell taronja. Té un cap molt gran amb

relació al cos, que indica que el cap és més important que la resta del cos a l'hora de sortir d'una crisi personal i també dona una estètica menys realista i amb encant, sense perdre la serietat del tema que tracta (vegeu Fig. 21 i Fig. 31 a la pàgina 41 pel resultat final).

3. És valent, ja que s'ha aventurat en les profunditats dels seus sentiments i aconsegueix afrontar els reptes que es troba. És el principal influenciador de la història.
4. S'escapa del cuc perquè no té cap possibilitat de guanyar una batalla contra ell. Es troba una aranya, que tot i que al principi li fa por s'acaben fent amics i l'ajuda a sortir del conjunt de coves.
5. És àgil i enginyós a l'hora de solucionar els reptes amb què es troba, però no domina la lluita cos a cos ni és capaç de saltar llargues distàncies.
6. Vol sortir del conjunt de coves, però està disposat a fer-hi amics durant el viatge.

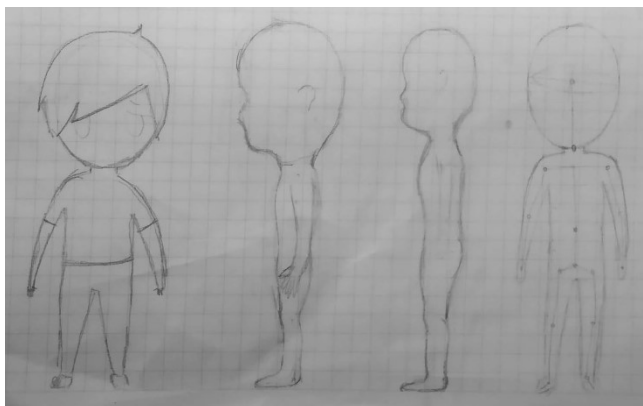


Fig. 21. Primers esbossos del disseny de l'Etho, a paper. Elaboració pròpia.

1. El següent personatge és l'aranya petita, el segon personatge més important en la història, tot i que, com que apareix més tard en la història, no està present a la versió Beta que s'ha publicat. Ajuda a l'Etho a sortir de la cova ja que hi ha zones en les quals ell sol no se n'hauria en sortit.

2. Comparada amb aranyes reals, és molt gran, 1'5m d'alçada. És de complexitat robusta, i negra en la seva majoria, tot i que són ben visibles unes marques blaves que desprenen certa claror (vegeu Fig. 22 i, pel model acabat, Fig.34 de la pàgina 42). Les seves potes li permeten escalar amb facilitat.



Fig. 22. Primers esbossos del disseny de l'Aranya, a paper. Elaboració pròpia.

Tot i que en un principi s'havien dissenyat les potes molt primes, tenia un realisme que no compartien la resta de dissenys així que es va modificar el disseny perquè les potes fossin més amples. Una vegada s'ha tingut clar com es volia que fos la seva estètica -però no es tenia el talent de dibuixar-la per a poder utilitzar les vistes de perfil i planta com a referència al crear el seu model 3D- en Néstor Gargallo, un amic dibuixant, va dibuixar diferents versions de les vistes (veure figura 23).

3. S'avorreix amb les seves germanes així que té ganes de fer amics i, per això, segueix l'Etho fins que ell l'acaba acceptant. També té ganes de viure aventures i sortir d'aquella cova. Li agraden molt els bolets que l'Etho recull, i això ajuda a reforçar la seva amistat.
4. Necessita emoció i aventura i no estar tancada a la cova, per això segueix l'Etho. Tot i que el noi li té por en un principi, ella l'ajuda i aconsegueix que siguin amics.
5. És bona escalant i pot fabricar teranyines, que els ajuden a sortir. No obstant, és impulsiva, i una vegada tirà l'Etho per un petit barranc per l'emoció de tenir els bolets.
6. L'emociona el sentiment de viure aventures i aconseguir els bolets que té l'Etho, ja que ella no pot arribar allà on creixen.

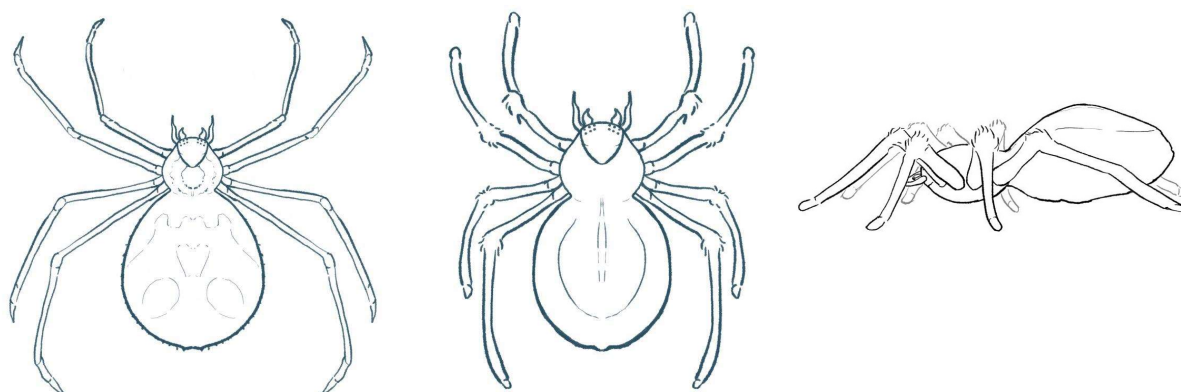


Fig. 23. Esbossos de Paranya, dibuix digital. Creats per N. Gargallo.

4.1.4 So

En aquest apartat s'explica breument en què ha consistit el disseny i desenvolupament de la música i els altres sons del videojoc.

En primer lloc, la música ha estat composta totalment per la Mar Álvarez. Ha agafat clara inspiració de Berlinist i les seves obres que apareixen en el videojoc Gris. És, en general, una música melancòlica, que acompanya la nostàlgia i soledat del protagonista. White Caves consta de tres cançons: una molt ambiental; una amb més ritme però també molt tranquil·la; i una molt tensa, que correspon a la persecució del cuc a l'Etho. La tensió d'aquesta cançó no es manté durant tota l'obra, sinó que quan el jugador arriba a la zona on aconsegueix el poder de plantar bolets, acaba amb un final psicodèlic.



Fig. 24. Gravació de les cançons, a l'esquerra en Darío i a la dreta la Mar. Elaboració pròpia.

Les tres peces musicals han estat compostes inicialment amb el Garage Band, però per obtenir una millor qualitat han estat gravades a l'estudi d'en Dario Nuviola (vegeu Fig. 24). El violí ha estat a càrrec de la Mar Álvarez, les veus han estat interpretades per l'Iris Marti i la Mar, la flauta per en Jaume Viñas, i en Dario Nuviola s'ha encarregat de la producció.

En segon lloc, malauradament, White Caves només té un efecte de so, la recollida de les "orbes" o punts. Quan el jugador n'aconsegueix es reproduceix un so de recompensa. Com que no es tracten de monedes, el so reproduït no és metàl·lic, sinó que és semblant a quelcom caient a l'aigua, amb bastant distorsió per donar un efecte positiu i de recompensa, per tal que el jugador sàpiga que ha recollit un objecte beneficiós. Aquest so ha estat descarregat gratuïtament de la pàgina storyblocks.com.

Tot i que es tenia la intenció d'afegir sons ambientals i sons en caminar, saltar, aterrar i altres accions, no se n'han afegit més. Això és perquè es va dedicar la major part del temps a la programació de les mecàniques i arreglar els múltiples problemes que hi havia. Afegir dos o tres sons més no m'hauria fet aprendre gaire, però programar noves mecàniques o corregir alguna d'existent permet que el resultat sigui millor i jo aprengui més.

4.2 Producció

Una vegada es van tenir relativament clars tots els aspectes relacionats amb el disseny del joc es va començar el seu desenvolupament.

4.2.1 Modelatge 3D i creació de textures

7.2.1.1. Definició i mètodes de modelatge

El modelatge 3D (en anglès *3D modeling*) és el procés utilitzat per a crear objectes tridimensionals virtualment utilitzant tecnologia computacional [40]. És comparable a esculpir models d'argila però en un ordinador. Es realitza a través de programes computacionals que s'encarreguen de les superfícies i els sòlids.

El procés de modelatge 3D es desenvolupa a partir de la representació matemàtica d'un objecte o element en la seva versió tridimensional en un espai virtual especialment dissenyat per aquesta funció [40]. Quan el model 3D està completat, es pot utilitzar per a altres espais virtuals (com videojocs) o en format físic fent ús de la tecnologia d'impressió 3D.

Hi ha molts mètodes diferents a l'hora de crear models 3D, però els dos utilitzats en aquest treball són els següents:

- *Subdivision/Box modeling*

És un dels mètodes més utilitzats de modelatge poligonal, on l'artista utilitza una forma geomètrica, com un cub, cilindre o esfera, i el va deformant i creant nous polígons fins a arribar a l'aparença desitjada [41]. Es comença amb una malla de poca resolució i després se li redefineix la forma. Posteriorment, es creen subdivisions de la malla per poder afegir detall (vegeu figura 25). Es van repetint els processos de redefinir i subdividir fins a aconseguir el detall desitjat.

- *Escultura digital*

És el millor mètode per a crear models orgànics, com personatges, animals, monstres o plantes. S'utilitzen tècniques molt semblants a les utilitzades a l'hora d'esculpir argila, però s'utilitza el ratolí i el teclat, i és molt més fàcil

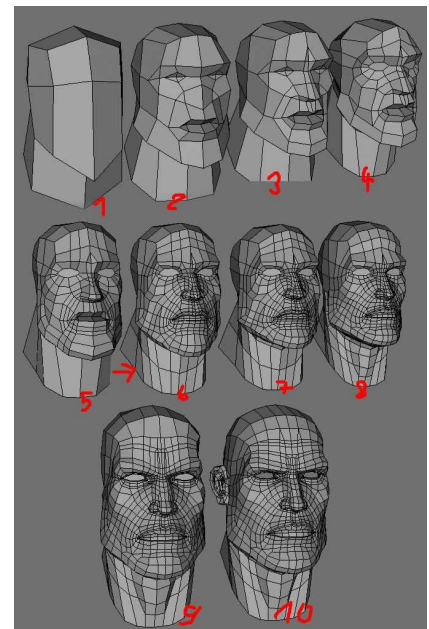


Fig. 25. Progressió d'un model 3D d'una cara, creat amb la tècnica de subdivisió o modelatge de caixa.

Extret de:

https://www.pinterest.es/pin/547046685965615431/?nic_v2=1a693YdwN

obtenir bons resultats. Generalment es comença amb una esfera i, amb l'ús de diferents pinzells, se li dona forma fins a assolir el detall desitjat (vegeu figura 26).

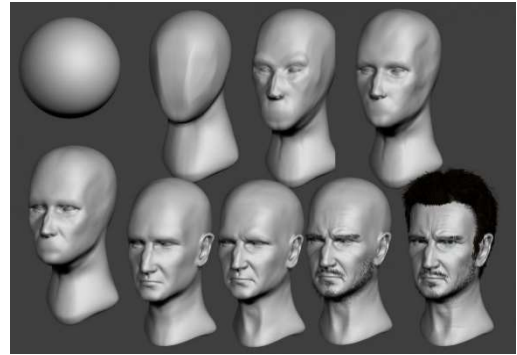


Fig. 26. Progressió d'un model 3D d'una cara, creat esculpint digitalment. Extret de: <https://www.tsumea.com/journal/150515/male-head-sculpt-01>

7.2.1.3. Models i textures de *White Caves*

El primer que s'ha fet a l'hora de començar a desenvolupar el videojoc han estat els models dels personatges i, posteriorment, del món i dels objectes. Es podria dir que els nou anys que porto jugant a Minecraft han ajudat, ja que en aquest joc s'ha centrat sobretot en els temes artístics i de construcció. A més a més, ja es tenia una mica d'experiència en l'àmbit del modelatge 3D perquè des de fa uns dos anys que se n'ha interessat i n'ha après pel seu compte. No obstant, mai s'ha agafat l'hàbit de modelar, sinó que es realitzava molt esporàdicament, alguna vegada cada cada mesos. Així doncs, s'ha començat amb el procés de modelatge amb una tècnica pràcticament nul·la.

A l'hora de saber quin programa aniria millor i com utilitzar-lo van ser molt útils certs vídeos de YouTube¹⁶, la majoria en anglès. En un inici es miraven vídeos tutorials, on ensenyen pas per pas què s'ha de fer, i normalment el resultat de l'aprenent acaba sent molt semblant i sense gaire innovació, però van bé per aprendre i interioritzar les opcions més bàsiques del programa. Una vegada es tingué més experiència i fluïdesa, s'han mirat vídeos que guien a l'espectador amb alguns passos, sí, però se salten els més bàsics i estan destinats a modeladors amb cert coneixement. A més a més, s'ha fet molt ús de fòrums en línia per mirar les preguntes dels altres usuaris i si s'encallava en algun moment, demanar la solució a modeladors més experts. Els més utilitzats són fòrums de Discord¹⁷, i els noms de les comunitats són "OpenVFX: The Blender Hub" i "Gabbitt 3D".

¹⁶ Youtube: Plataforma en línia per a mirar vídeos a la carta que altres usuaris han penjat.

¹⁷ Discord: Plataforma en línia de xat a temps real, on els usuaris poden crear comunitats i unir-se a existents. Els temes d'aquestes són molt variats, com videojocs, matemàtiques, ciència, llengües, viatges, sèries i grup d'amics.

El programa utilitzat va ser Blender, un programa de modelatge 3D, animació 3D i texturització gratuït. És molt complet i *Open Source*, així que és una de les opcions més utilitzades tant per aprenents com per professionals.

Els models 3D acostumen a estar formats per triangles, i generalment quan més petits més detall té el model (veure figura 27) però l'ordinador consumeix més recursos. Moltes vegades es busca l'equilibri entre pocs polígons però un model bonic.

L'estètica de *White Caves* és *Low Poly*, és a dir, els models estan formats per molt pocs polígons. De vegades aquest recurs és utilitzat quan no es vol invertir molt de temps en models poc importants, com els que es troben al fons d'una escena que no necessiten detall, però també és molt utilitzat com a recurs artístic (vegeu figura 28).

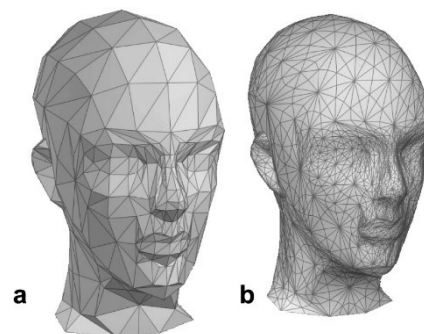


Fig. 27. Comparació entre un model amb pocs polígons (A) i un amb molts (B). Extret de: <https://kieranpbrown.wordpress.com/what-are-polygons-in-3d-modelling/>

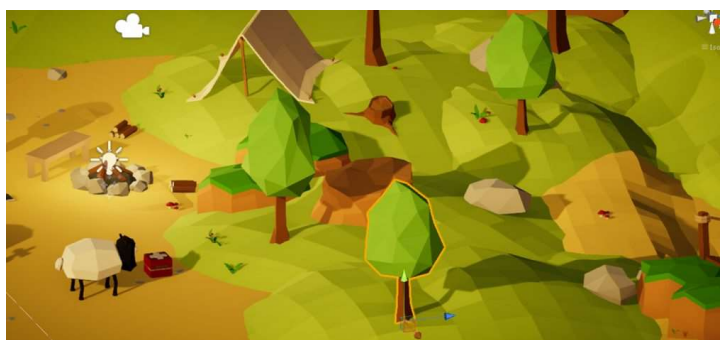


Fig. 28. Escena 3D composta de models low poly, que donen una estètica molt bonica en algunes ocasions. Extret de: <https://www.blendernation.com/2018/05/01/low-poly-blender-tree-wind-effect-in-unity/>

Etho

El primer model que es va crear és el del protagonista, Etho. Ja es tenia més o menys clara l'estètica del personatge (vegeu figura 21 a la pàgina 35) però el model 3D va passar per diferents versions (vegeu figures 28 i 29). Per començar a agafar fluïdesa amb el programa, es va seguir al peu de la lletra un tutorial de Youtube anomenat "How to Create a Low Poly Character in Blender 2.8" i creat per l'usuari CG Geek. Després, Es va retocar el model fins a deixar-lo al gust.

La textura està creada a partir de colors únics i sense detall (vegeu Fig. 31), com és típic en l'estètica Low Poly.

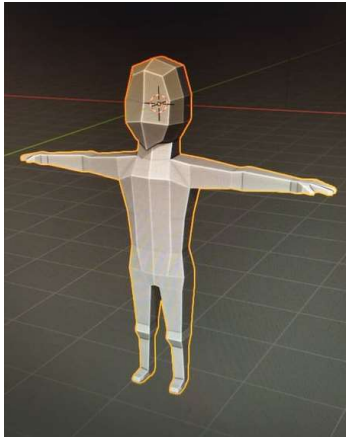


Fig. 28. Model inicial de l'Etho. Elaboració pròpia.

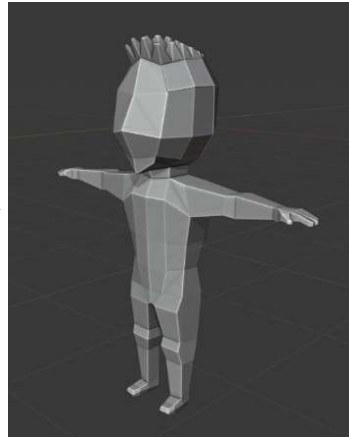


Fig. 30. Model 3D final d'Etho. Elaboració



Fig. 31. Model i textura definitiva d'Etho. Elaboració pròpia.

Aranya

La creació del model de l'aranya ha estat molt diferent que el d'Etho. En primer lloc, es va fer recerca sobre l'anatomia de les aranyes i les diferents classes d'aspectes que existeixen. Es va decidir realitzar una aranya formosa i amb potes robustes. El disseny a paper va començar amb l'aspecte d'una formiga, després es van dibuixar unes quantes versions (vegeu figura 22 a la pàgina 35) més i finalment en Néstor Gargallo, dibuixant, va dissenyar diversos aspectes d'aranyes perquè utilitzar-los com a referència al modelar (vegeu figura 23 a la pàgina 36).

Quan l'estètica de l'aranya estava clara, es va començar amb el modelatge. Es va utilitzar, de nou, la tècnica de subdivisió, ja que es volia un resultat low poly, té una forma difícil d'aconseguir esculpint i no requereix gaire detall. Es va començar amb dues esferes estretes pels pols, que corresponen a l'abdomen i al cefalotòrax. Després, es van extreure polígons que corresponen a les potes i, finalment, es van crear els quelícers i els pedipalps.

El model provisional (veure figura 32) va ser creat molt ràpidament, en unes dues hores. No obstant, el resultat no va ser complaent ja que no es corresponia a l'estètica de videojoc a la que s'aspirava. El problema principal era que les potes eren massa primes amb relació a la seva longitud i el cos, així que poc després vaig fer les extremitats més robustes (veure figura 33).



Fig. 32. Model i textura provisional de l'aranya. Elaboració pròpia.



Fig. 33. Model definitiu de l'aranya. Elaboració pròpia.



Fig. 34. Model i textura definitiva de l'aranya. Elaboració pròpia.

Pel que fa a les textures, en un inici es van crear unes textures amb més detall que les que acabarien sent definitives però bastant lletges. A més a més, per desconexença de la correcta utilització del programa, les textures se sobreposaren (vegeu foto 32), així que més tard es va crear una nova textura des de zero, ara sí, utilitzant correctament el programari. El resultat, tot i que és més simple, és més bonic i correspon a l'estètica buscada (vegeu foto 34).

En visualitzar les fotos s'ha de tenir en compte que no són *renders*, sinó captures de pantalla del programa. Així doncs, no s'ha de tenir en compte la il·luminació, les ombres ni el material dels models, ja que només són vistes prèvies del resultat final sense la il·luminació correcta.

Cuc gegant

A l'hora de dissenyar l'estètica del cuc es va començar amb una recerca a Google sobre altres exemples de cucs gegants. Després es van dibuixar esbossos fins a aconseguir-ne un de bo (vegeu Fig. 35). La part més difícil de dissenyar va ser la part del final del cuc; no se sabia com acabar-la. Inicialment es va pensar en fer dues punxes, però finalment es va decidir acabar el cuc amb una bola blava al final. La intenció era

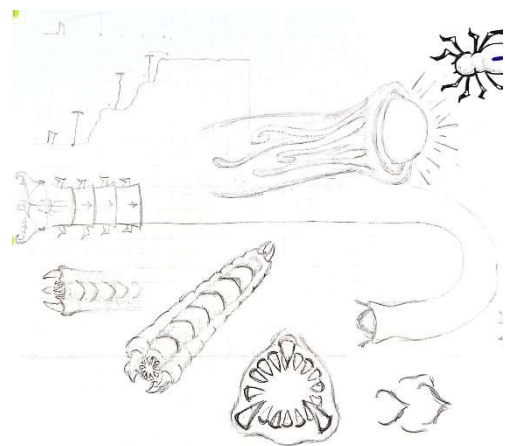


Fig. 35. Esbossos del disseny del cuc gegant. Elaboració pròpia.

que aquesta esfera emetés llum i que, potser, el jugador podria matar el cuc donant-li un fort cop a l'esfera. Al final, no es va dur a terme cap d'aquestes dues idees.

Pel que fa al modelatge del cuc, es van utilitzar les dues tècniques explicades anteriorment. Es va començar el model a partir d'un cilindre, que es va subdividir i editar amb la tècnica de *box modeling* per obtenir la forma bàsica del cuc (Fig. 36). A continuació, es va esculpir el cuc per donar-li més detall. Es va afegir la boca, la cua i relleus que simulaven escates i venes (Fig. 37 i 38). També es van crear les dents, de nou, amb la tècnica de *box modeling*. Les escates clarament no donaven un bon resultat així que es van treure.

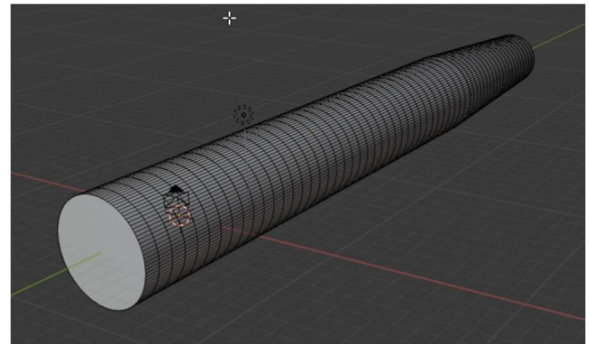


Fig. 36. Model primitiu del cuc gegant. Elaboració pròpia.

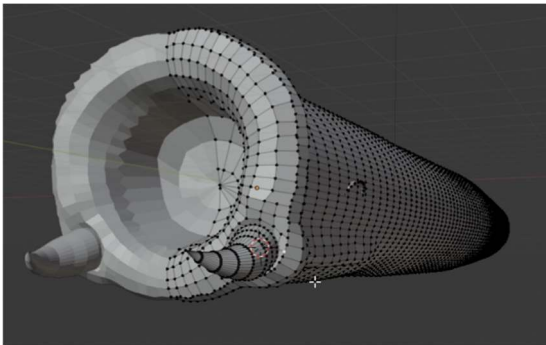


Fig. 37. Cuc gegant; començament d'esculpir. Elaboració pròpia.

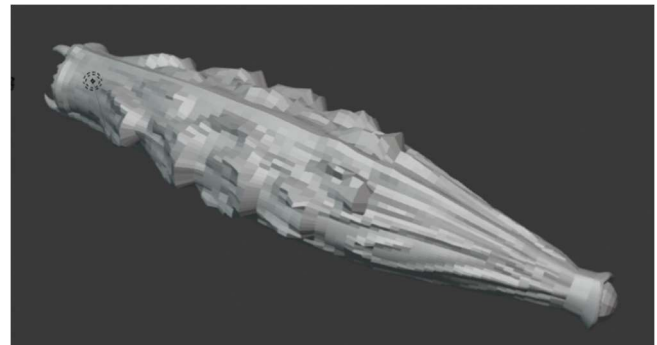


Fig. 38. Cuc gegant; amb escates provisionals. Elaboració pròpia.

A continuació es va començar amb la creació de la textura. Va ser un procés difícil que va patir diverses versions. El primer intent (Fig. 39) queda molt malament al joc, i seria més típic d'una estètica com els jocs de Mario en 3D. A l'hora de crear aquesta textura encara es tenia la intenció de fer-li escates, però és clar que el resultat no era bo. També es van pintar de blau cel les venes, simulant línies de poder que venen des de l'esfera final.

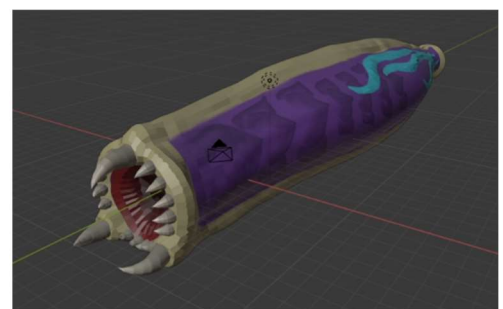


Fig. 39. Textura provisional del cuc gegant. Elaboració pròpia.

Finalment, es va crear una nova textura més senzilla però amb millors colors (Fig. 40 i 41). Les dents, l'esfera i la panxa tenen tons difuminats, tot i que en el videojoc no s'aprecien gens. La decisió de quin color utilitzar va ser difícil, ja que si fos marró, el color més comú per aquests cucs, no ressaltaria amb el fons de l'escenari, també marró perquè és una cova.



Fig. 40. Model i textura final del cuc gegant.
Elaboració pròpia.



Fig. 41. Model i textura final del cuc gegant.
Elaboració pròpia.

Món

La creació del model 3D i textura del món ha estat un procés llarg amb molts errors fruitos de la inexperiència que m'han enriquit com a modelador.

Ja que no s'havia creat cap model tan gran i no es sabia per on començar, es va utilitzar el mateix mètode que va utilitzar Davide Prestino en el seu *timelapse* penjat a YouTube¹⁸.

Així doncs, per començar es va crear un model primitiu amb molt pocs polígons amb la tècnica de *box modeling* (vegeu Fig. 42), seguint la forma del disseny del món (vegeu Fig. 20 a la pàgina 33). A l'hora de crear el model primitiu del món es va utilitzar el model ja creat de l'Etho com a referència per saber les dimensions que havia de tenir el món. Tot i així, era molt difícil controlar l'escala de la mida, així que hi ha zones que van sortir massa grans i d'altres massa petites.

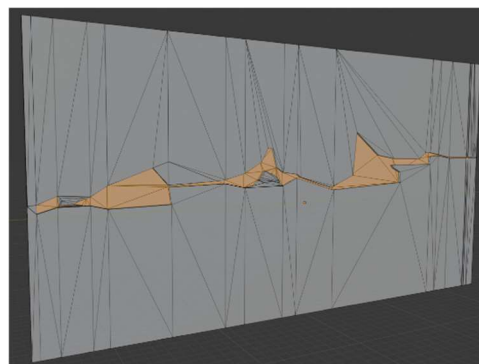


Fig. 42. Model primitiu del món. Les cares taronges corresponen al que esdevindrà la cova i els camins. Té 844 polígons. Elaboració pròpia.

¹⁸ Timelapse penjat per Davide Prestino. Títol: "Design a Game Level 3D Timelapse in Blender 2.8 | Blender Lowpoly". URL: <https://youtu.be/-y04xne1jm8>

A continuació, es va dur a terme l'escultura digital per millorar els detalls del model. Al principi, es progressava molt lentament perquè els polígons es desenganxaven entre ells. Reajuntar els triangles és un procés llarg i molt avorrit. No obstant, quan ja es van enganjar correctament tots els polígons de nou, esculpir el món va ser una de les parts més divertides del treball.

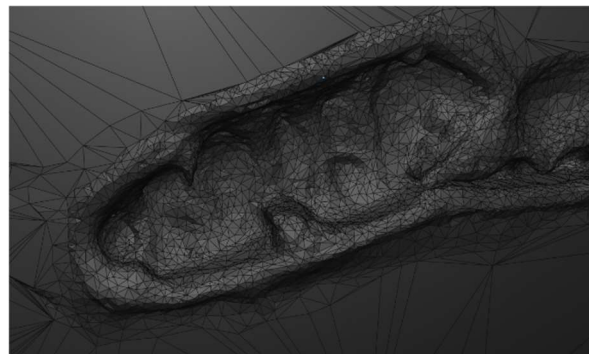


Fig. 43. Primera sala del joc. Elaboració pròpia.

Primer es van esculpir les cantonades, per tal que no quedés un angle recte. La primera zona, allà on apareix el jugador al començar el joc (vegeu Fig. 43), va quedar més malament que les altres zones, ja que encara no s'havia agafat pràctica a l'escultura digital. També es van arrodonir totes les cares visibles pel jugador, sense afegir detall.

Després, es va afegir complexitat a les zones, procurant que cada sala tingués la seva especialitat, com el característic centre de la segona sala (vegeu Fig. 44). També es van afegir petits detalls com estalactites i estalagmites, petites coves secundàries o conjunts de pedres que bloquegen el camí (vegeu Fig. 45).

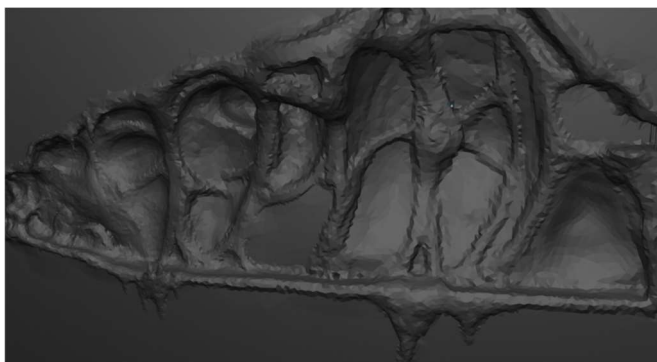


Fig. 44. Model de la segona sala. Elaboració pròpia.

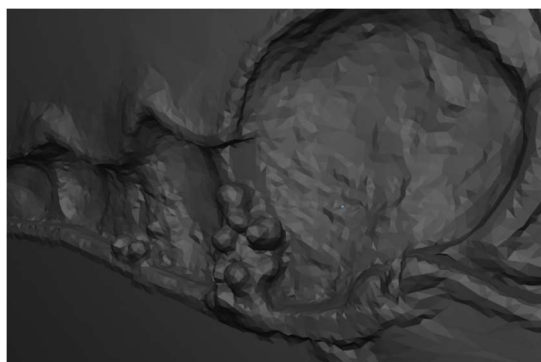


Fig. 45. Model de la tercera sala. Un cúmulo de pedres barra el camí. Elaboració pròpia.

En acabar el procés de modelatge, el model tenia uns 800.000 polígons, així que, perquè pesés menys i sobretot el rendiment del joc fos molt millor, van ser reduïts a 140.000. Al començar a reduir-los es va fer manualment, que comporta molt de temps, un resultat dolent i realment el nombre de triangles es redueix poc. A més a més, es va fer l'error d'apropar massa els límits del món a la part de la cova visible, així que en algunes zones del joc, si el jugador té el zoom al mínim pot ser que vegi els contorns del model del món, i queda molt malament.

Per sort, es va trobar un mètode molt més ràpid i que dona millors resultats a l'hora de reduir el nombre de polígons. Blender et permet triangular qualsevol model 3D molt fàcilment gràcies a un *modifier*. A l'opció de

“Ratio” es tria com de molt s’ha de triangular i reduir el model, sent 1 el model original sense alterar i 0 que es quedi sense cap polígon. D’altra banda, aquesta opció també és molt recomanada perquè, sempre que pot, redueix els polígons redundants. És a dir, redueix dràsticament el nombre de triangles de les zones planes, perquè amb pocs triangles ja es pot assolir una superfície plana. En canvi, redueix menys les zones que tinguin bastant detall, perquè el model no perdi qualitat.

A continuació s’ensenya un exemple de triangulació a la sala prèviament ensenyada (Fig. 44), aplicant-li un Ràtio de 0.1 (vegeu Fig. 46). El nombre total de polígons s’ha reduït de 140.000 a 40.000 així que, com es pot observar, el nou model té uns triangles molt més grans que l’anterior.

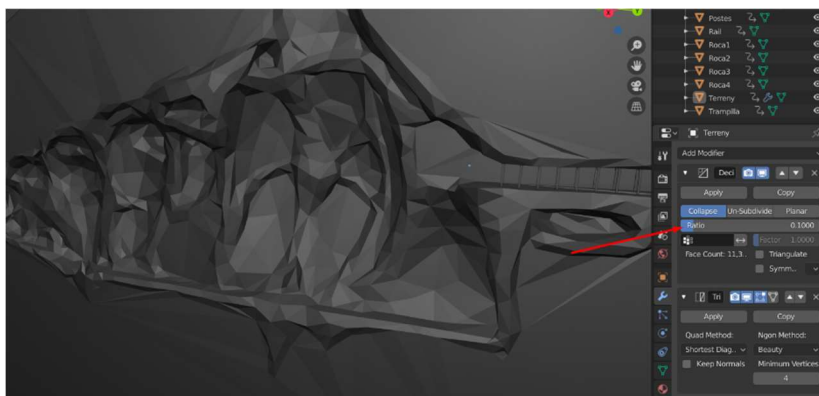


Fig. 46. Reductor de polígons a partir de triangulació. Exemple utilitzant la segona sala. La fletxa senyala la opció que s’ha d’alterar per augmentar o disminuir els polígons. Elaboració pròpia.



Fig. 47. Textura final del món. Quarta sala. Elaboració pròpia.

La creació de textures va ser un procediment molt lent. Des d’un principi es van escollir colors que no quedaven bé, així que es van canviar un total de tres vegades. El resultat final (Fig. 47 i 48) consisteix en colors marrons, grisos i algun to groguenc, amb molts difuminats. Vegeu L’Annex 2 per a veure les fotografies detallades del món complet.

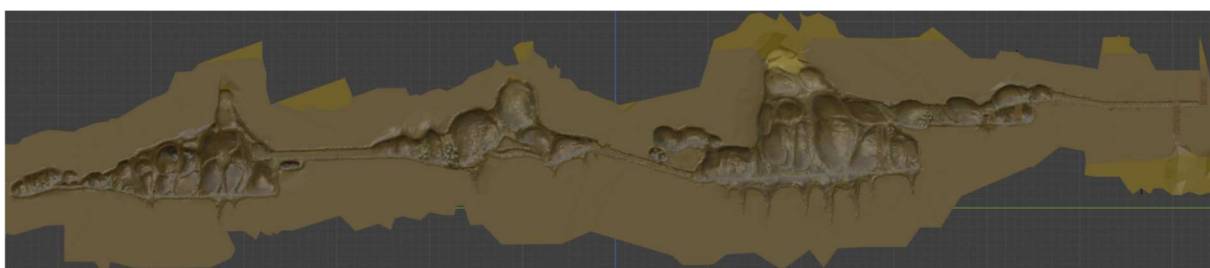


Fig. 48. Model complet del món. La il·luminació i els reflexes no són els que es veuen en el videojoc. Elaboració pròpia.

4.2.2 Animació 3D

L'animació 3D és el procés d'animar objectes que apareixen en un espai de tres dimensions [42]. Poden ser girats i moguts com objectes reals. La majoria d'animacions cobren vida quan el que es mou no és només l'objecte en si, sinó els seus ossos o, en anglès, *bones*. Aquests ossos normalment corresponen a un os real, però no en tots els casos. Serveixen per dictar com s'ha de comportar aquella zona del model, perquè cada os té associats certs polígons. Per exemple, en la figura 49, es poden veure els diferents ossos del model de l'Etho, representats amb octaedres. En la foto està seleccionat l'os de la cuixa dreta, així que els polígons més pròxims a aquest són vermells (l'os els controla totalment), mentre que els de color blau no estan afectats per l'os seleccionat.

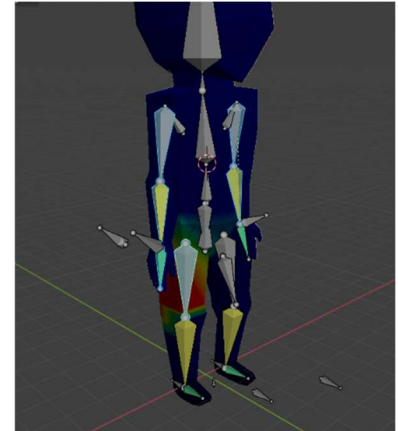


Fig. 49. **Bones de l'Etho.** Els polígons vermells són els més afectats per l'os de la cuixa dreta. *Elaboració pròpia.*

Blender, el programa que es va utilitzar per fer el modelatge 3D i la creació de textures, també incorpora una bona interfície per a crear animacions (vegeu Fig. 50). En aquesta foto els ossos no estan representats com a octaedres, sinó que la seva forma depèn de la seva funció. Per exemple, els cubs molt allargats són els ossos que modifiquen els polígons en si, però l'animador no pot moure'ls a mà. Per moure'ls ha d'utilitzar les esferes, ja que en aquest *rig*¹⁹ s'utilitza una tècnica anomenada Inverse Kinematics per animar les potes de l'aranya. Aquesta és una tècnica que també s'utilitza en la creació dels moviments dels braços robòtics perquè, al tenir diversos ossos articulats, per l'animador és més fàcil moure únicament la mà, i que els ossos superior i inferior del braç es moguin sols. Els petits cubs serveixen com a marcadors de direcció del que correspondria al colze. El model de l'Etho (Fig. 49) també utilitza aquesta tècnica, però les formes són totes iguals perquè és un rig molt senzill.

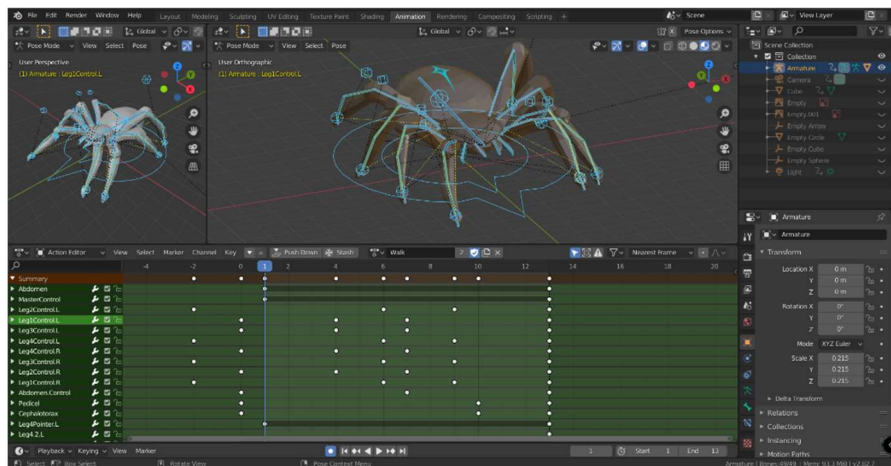


Fig. 50. **Interfície d'animació a Blender.** Rig de l'aranya. *Elaboració pròpia.*

¹⁹ Rig: l'esquelet d'un model 3D perquè pugui ser animat. Sense un rig, el model es pot moure, però no deformar (és necessari, per exemple, perquè el model d'una persona pugui flexionar els braços o rotar el cap).

Per aprendre a utilitzar la interfície d'animació es van veure els 7 vídeos de YouTube de Grant Abbit per aprendre a animar²⁰ i seguir els seus consells. D'altra banda, també es van veure vídeos puntuals d'altres creadors com CG Geek o Royal Skies LLC. Tal i com va aconsellar Grant Abbit, es va començar a aprendre a animar utilitzant formes bàsiques com cubs o esferes. Una vegada ja se sabia utilitzar bé el programa, es va crear el rig de l'Etho per, posteriorment, animar-lo. Es va fer el mateix per l'aranya i el cuc gegant.

Les animacions més treballades van ser les del personatge, ja que en té moltes i són les que es veuen des de més a prop i durant més estona (vegeu Fig. 51). Aquesta taula és un llistat de totes les animacions del personatge, amb els fotogrames on comença i acaba. La majoria d'aquestes animacions són cícliques, és a dir, quan s'acaba es torna a reproduir si no té ordre de parar. Per tant, l'últim fotograma i el primer han de ser pràcticament iguals. Malauradament, l'animació de "walk", "surprise", "surface swim", "underwater swim" no es van utilitzar a la versió Beta de White Caves, perquè corresponen a accions que es duen a terme més endavant al joc.

Etho	Begins	Ends
Walk:	1	31
Jump:	40	65
Sneak:	70	109
Run:	120	146
Idle	250	2500
Surprise	2550	2562
ClimbLadder	2580	2629
DescentLadder	2670	2719
SurfaceSwim	2800	2845
UnderwaterSwim	2900	2945

Fig. 51. Temps que dura cada animació de l'Etho, en fotogrames. Elaboració pròpia.

Per crear l'animació de caminar es va utilitzar de referència la Figura 52, extreta d'un vídeo d'Alan Becker. Primer es van posicionar els ossos perquè l'Etho estigués en la posició de "contact" en el primer fotograma, i a la posició de "passing" en el vuitè. Després, en el quart i dotzè fotograma es va crear la posició de "down" i "up" respectivament. El

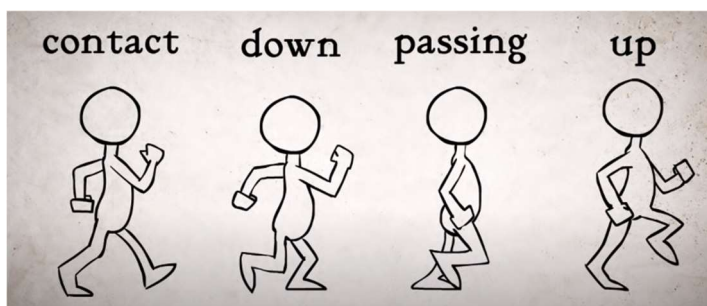


Fig. 52. Keyframes de l'animació de caminar. Extret de: <https://youtu.be/2y6aVz0Acx0>.

programa crea els fotogrames restants tal que l'animació quedi el màxim fluida, així que només va ser necessari corregir alguna de les posicions establertes pel programa que no quedaven bé.

²⁰ Conjunt de tutorials penjats per Grant Abbit. Títol: "Beginners guide to animation in 2.8". URL: <https://www.youtube.com/playlist?list=PLn3ukorJv4vvHr6RMoXrZSMVqmOKlqbBR>

4.2.3 Programació

En aquest apartat del treball s'explica breument la programació d'alguna de les mecàniques i funcions del joc, i s'ensenya la part rellevant del codi i una explicació resumida de la seva funció. No és necessari cap coneixement sobre programació per a comprendre aquest apartat, ja que no parlaré de la sintaxi del codi, sinó únicament de la seva semàntica. No es pretén que el lector sigui capaç de programar un videojoc després de llegir aquest text, ja hi ha molt bons manuals a Internet.

No s'inclouen les línies de codi de declaració de variables, llibreries utilitzades o altres parts del codi que probablement no són d'interès per la major part del públic. Únicament les més interessants, ja que no es poden explicar totes perquè n'hi ha més de 1350.

Etho

El primer que s'ha escrit a l'hora de programar el codi que controla el personatge Etho ha estat el seu moviment i, per això, es necessita saber amb precisió què és el que el jugador desitja fer. Així doncs, aquest codi s'encarrega de llegir les instruccions que dona el jugador.

```
1. void Update()
2.     {
3.         if(moveEnabled)
4.             {
5.                 horizontalMove = Input.GetAxisRaw("Horizontal") * runSpeed;
6.
7.                 if (Input.GetButtonDown("Jump")) jump = true;
8.
9.                 if (Input.GetButtonDown("Crouch")) crouch = true;
10.                else if (Input.GetButtonUp("Crouch")) crouch = false;
11.
12.                if (rb.velocity.y < 4)
13.                    {
14.                        rb.velocity += Vector3.up * Physics.gravity.y * (fallMultiplier - 1) * Time.deltaTime;
15.                    }
16.                else if (rb.velocity.y > 0 && !Input.GetButton("Jump"))
17.                    {
18.                        rb.velocity += Vector3.up * Physics.gravity.y * (lowJumpMultiplier - 1) * Time.deltaTime;
19.                    }
20.            }
21.    }
```

Aquest codi està dins del mètode `Update()`, així que serà executat cada fotograma si el primer condicional es compleix (si el moviment està activat).

En primer lloc, a la **Línia 5**, es llegeix si el jugador està prement els botons associats al moviment horitzontal (anomenats “Horizontal” i que per predeterminat són la tecla A i la D). Aquest valor (-1 per A o esquerra, 1 per D o dreta i 0 per cap tecla premuda) és multiplicat a la velocitat que es vol que es mogui el jugador. Així doncs, si vol anar cap a l’esquerra, tindrà una velocitat negativa, i si vol anar a la dreta, serà positiva. Si no es prem cap tecla o es premen les dues, la velocitat serà zero. Aquest valor és associat a la variable `horizontalMove` per ser utilitzat més endavant.

A continuació, a la **Línia 7**, si el jugador prem el botó associat a saltar (anomenat “Jump”), la variable `jump` és positiva i, si no, és negativa. El mateix succeeix amb el moviment per gatejar (anomenat “Crouch”) a la **Línia 9 i 10**.

Per últim s’altera la gravetat del jugador. Com que White Caves està creat amb Unity, una Game Engine amb sistema de físiques, no cal programar la seva gravetat, sinó que es pot editar directament des del programa. En aquest cas, les **línies 12-19** editen la gravetat estipulada prèviament per Unity per obtenir un millor resultat i control:

D’una banda, en la majoria de jocs de plataforma el salt no és realista, no triga el mateix en pujar que en baixar, sinó que baixa més ràpidament (veure Fig. 53).



Fig. 53. Salt de Mario a SuperMario. Al pujar triga 18 fotogrames, però al baixar només 10. Extret de: <https://youtu.be/7KiK0Aqtmzc>.

Aquest resultat es pot obtenir de diferents formes, però en aquest cas es controla la velocitat alterant la gravetat. Així doncs, amb un condicional en la **Línia 12** es detecta si la velocitat vertical del jugador és inferior a 4 (quan comença a baixar) i, en cas que sí, en la **Línia 14** es modifica la seva gravetat perquè sigui més gran que la predeterminada i caigui més ràpidament.

De l’altra banda, es va decidir que el jugador pogués controlar l’altura del salt segons quanta estona mantingués premut el botó de saltar. Com que no es tenia clara aquesta decisió, es va realitzar una enquesta per Instagram, on clarament va quedar clar que es prefereix que el jugador pugui controlar l’altura del salt (vegeu Fig. 54). Perquè l’altura fos controlable i quedés un moviment fluid es va fer el següent: primer, **Línia 16**, amb un condicional es mira quan el jugador està pujant (velocitat vertical positiva) però ja no està prement el botó de saltar. Si és així, en la **Línia 18**, s’augmenta la seva gravetat perquè comenci a caure.



Fig. 54. Enquesta a Instagram sobre si es prefereix que el jugador pugui controlar l’altura del salt. Elaboració pròpia.


```

1. public void Move(float move, bool crouch, bool jump)
2.     {
3.         if (!crouch)
4.         {
5.             // Si el personatge té un sostre que l'impedeix posar-se dempeus, que es
mantingui gatejant
6.             if (Physics2D.OverlapCircle(m_CeilingCheck.position, k_CeilingRadius, m_Wh
atIsGround))
7.             {
8.                 crouch = true;
9.             }
10.        }
11.
12.        if (isGrounded || m_AirControl)
13.        {
14.            if (crouch)
15.            {
16.                // Reduir la velocitat
17.                move *= m_CrouchSpeed;
18.
19.                // Desactivar un dels colliders
20.                if (m_CrouchDisableCollider != null)
                m_CrouchDisableCollider.enabled = false;
21.            }
22.            else
23.            {
24.                // Activar el collider
25.                if (m_CrouchDisableCollider != null)
                m_CrouchDisableCollider.enabled = true;
26.            }
27.
28.            // Moure el jugador a la velocitat desitjada
29.            Vector3 targetVelocity = new Vector2(move * 10f, m_Rigidbody.velocity.y);
30.            m_Rigidbody.velocity = Vector3.SmoothDamp(m_Rigidbody.velocity, targetVelo
city, ref velocity, m_MovementSmoothing);
31.
32.            if (move > 0 && !m_FacingRight) Flip();
33.            else if (move < 0 && m_FacingRight) Flip();
34.        }
35.
36.        // Si el jugador ha de saltar
37.        if (isGrounded && jump && !ladder.detected)
38.        {
39.            isGrounded = false;
40.            m_Rigidbody.AddForce(new Vector3(0f, m_JumpForce, 0f), ForceMode.VelocityC
hange);
41.        }
42.    }

```

Aquest codi és el que executa els principals moviments de l'Etho, i es divideix en tres parts: gatejar, caminar i saltar.

- En primer lloc, el model de l'Etho està format per dos *colliders* (detectors de col·lisions) diferents, un als peus i cames i l'altre a la resta del cos (vegeu Fig. 55). La col·lisió d'aquest model 3D no es calcula a partir dels seus polígons, ja que gastaria molts més recursos de l'ordinador. En altres paraules, el model 3D i les textures de l'Etho només tenen una funció visual.

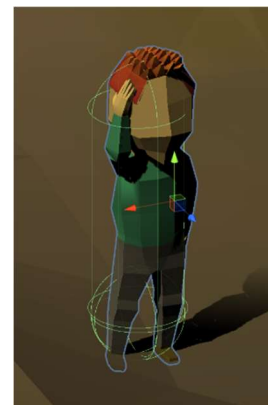


Fig. 55. *Colliders* de l'Etho. Els dos detectors de col·lisions estan representats per línies verdes. Elaboració pròpia.

Quan està gatejant representa que el personatge està estirat a terra, així que hauria de poder passar per forats petits. Per aconseguir això es desactiva el *collider* superior a la **Línia 20** i, si es deixa de gatejar, es torna a activar a la **Línia 25**. No obstant, si el personatge encara està en el forat i, doncs, té un sostre just sobre seu, no s'ha de poder aixecar, ja que no hi cabria i es posaria dins la paret. Per evitar-ho, es mira l'estat de la variable *crouch*, explicada anteriorment. Si és negativa, és a dir, el jugador no està prement el botó de gatejar, a la **Línia 6** es comprova que el *collider* superior no estigui dins la paret. Si dona positiu, *crouch* es torna a posar en positiu (**Línia 8**), ja que encara no es pot aixecar.

move és una variable determinada per *horizontalMove*, explicada anteriorment, i que determina la velocitat horitzontal del jugador. Quan gateja, com que ha d'anar més lent, a la **Línia 17** es multiplica *move* per *m_CrouchSpeed*, que correspon a un número entre 0 i 1, per tal de disminuir la seva velocitat.

- En segon lloc, s'ha de moure el jugador horitzontalment. Per fer-ho, primer es calcula la velocitat a què ha d'anar a la **Línia 29** i, després, a la següent línia, s'aplica aquesta velocitat al personatge, aplicant-hi un efecte per fer el moviment més suau. A continuació, si el jugador ha d'anar cap a la dreta, però està mirant cap a l'esquerra, se'l gira, i viceversa (**Línies 32 i 33**). Per girar-lo, s'utilitza un *method*, anomenat *Flip()*, que es troba en una altra part del codi per fer la lectura més neta. La funció de *Flip()* és editar l'escala horitzontal del personatge per -1, és a dir, canvia la direcció en què mira emmirallant-lo (també es podria fer amb una rotació de 180°, però el resultat és pitjor).
- Finalment, a l'hora de fer saltar el jugador, es detecten tres variables. *isGrounded* és veritable quan el personatge està tocant a terra, i falsa quan no (explicada amb més detall més endavant). *jump*, com s'ha explicat anteriorment, és veritable quan el jugador prem la tecla de saltar, i falsa quan no. *ladder.detected* correspon a la part final del joc i no al moviment del personatge en sí, i serveix per evitar que l'Etho salti quan està pujant o baixant una escala.

Si el jugador està tocant a terra, està prement “saltar” i no està en una escala, s’aplica una força vertical al personatge per tal que es mogui verticalment cap amunt, simulant un salt.

Quant a la detecció de si el personatge està tocant el terra hi ha molts mètodes possibles. Va ser una de les primeres funcions que es van programar i, per la falta d’experiència i coneixement, va ser molt difícil. Cada mètode té els seus avantatges i inconvenients. En aquest cas, es va utilitzar el següent:

```
1. void GroundCheck()
2.     {
3.         if (Physics.Raycast(transform.position, Vector3.down, distToGround)) isGrounded = true;
4.         else isGrounded = false;
5.     }
```

A la **Línia 3**, hi ha un condicional que detecta si un raig que surt des de la posició del jugador cap a baix i de longitud `distToGround` xoca amb qualsevol *collider*. Si es produeix la col·lisió vol dir que el personatge està tocant a terra; d'altra forma, no (**Línia 4**).

Per acabar amb el codi de l’Etho s’explica el codi que controla la reproducció de les seves animacions:

```
1. void Animations()
2.     {
3.         //set the animation to idle
4.         if (!Input.GetButton("Horizontal") && !Input.GetButton("Jump") && !Input.GetButton("Crouch") && !ladder)
5.         {
6.             anim.SetInteger("condition", 0);
7.         }
8.         //set the animation for running
9.         if (Input.GetButton("Horizontal") && (isGrounded || m_AirControl))
10.        {
11.            anim.SetInteger("condition", 1);
12.        }
13.        //set the animation for jumping
14.        if (Input.GetButton("Jump") && isGrounded && !ladder.detected)
15.        {
16.            anim.SetInteger("condition", 2);
17.        }
18.        //set the animation for crouching
19.        if (Input.GetButton("Crouch") && isGrounded && !ladder.detected)
20.        {
21.            anim.SetInteger("condition", 3);
22.        }
```

```

23.     //set the animation to descent ladder
24.     if (ladder.detected && Input.GetAxisRaw("Vertical") == -1)
25.     {
26.         anim.SetInteger("condition", 5);
27.     }
28.     //set the animation to ascent ladder
29.     if (ladder.detected && Input.GetAxisRaw("Vertical") == 1)
30.     {
31.         anim.SetInteger("condition", 4);
32.     }
33.
34.     }

```

En resum, la forma utilitzada per canviar les animacions és la de canviar la variable `condition` a diferents valors. Des de Unity, aquests valors són processats i, segons la programació que hagi preparat el desenvolupador prèviament, es reproduirà una animació o altra. Per exemple, si `condition == 0`, es reproduïx l'animació de pausa, que correspon a quan l'usuari no està prement cap tecla i el personatge fa petits moviments com respirar, moure el cap, canviar el pes de cama o saludar a la càmera; si `condition == 1` es reproduïx l'animació de córrer; si és 2, la de saltar; si és 3 la de gatejar, etc.

Cuc gegant

Quant al cuc gegant, s'explica la detecció del seu moviment i intel·ligència artificial bàsica.

```

1. private void Update()
2.     {
3.         float distX2 = player.position.x - transform.position.x;
4.         if (distX2 - 100 > 0) start = true;
5.
6.         if (start) //només s'executa si Etho ha estat a la dreta del cuc
7.         {
8.             if (!played)
9.             {
10.                 played = true;
11.                 FindObjectOfType<AudioManager>().playMusic2 = true;
12.             }
13.
14.             //distància del cuc al jugador:
15.             distToPlayer = Vector3.Distance(transform.position, player.position);
16.             distY = Math.Abs(transform.position.y - player.position.y);
17.             distX = Math.Abs(transform.position.x - player.position.x);
18.
19.             if (distY < maxYdist)

```

```

20.         {
21.             if (distToPlayer < agroRange && distY < sameYRange && distToPlayer > sameYRange && (!v_HasRun || v_Detected))
22.                 {
23.                     ChasePlayer();
24.                 }
25.             else
26.                 {
27.                     StopChasingPlayer();
28.                 }
29.         }
30.     else
31.     {
32.         if (distToPlayer < agroRange && ((distX > minAgroRange && !isFollowing) || (distX > attackRange && isFollowing)))
33.             {
34.                 ChasePlayer();
35.             }
36.         else
37.             {
38.                 StopChasingPlayer();
39.             }
40.         }
41.     }
42. }
43.
44. private void ChasePlayer()
45.     {
46.         isFollowing = true;
47.         anim.SetInteger("condition", 0);
48.         if(transform.position.x < player.position.x)
49.             {
50.                 transform.Translate(0, 0, moveSpeed * Time.deltaTime); //el cuc està a l'esquerra així que s'ha de moure a la dreta
51.                 transform.localScale = new Vector3(1.5f, 1.5f, 1.5f);
52.             }
53.         else if (transform.position.x > player.position.x)
54.             {
55.                 transform.Translate(0, 0, -moveSpeed * Time.deltaTime); //el cuc està a la dreta així que s'ha de moure a l'esquerra
56.                 transform.localScale = new Vector3(1.5f, 1.5f, -1.5f);
57.             }
58.         }
59.
60.     private void StopChasingPlayer()
61.     {
62.         isFollowing = false;
63.         anim.SetInteger("condition", 1);
64.     }

```

Aquest codi correspon al moviment del cuc. L'animal està adormit fins que l'Etho li passa per sota i, quan el personatge està 100u més a la dreta del cuc, aquest es desperta i comença a perseguir el jugador. Així doncs, només es vol que el moviment del cuc estigui actiu si el personatge ha estat una vegada 100u cap a la seva dreta (**Línies 3, 4 i 6**).

A la **Línia 11** s'executa la cançó tensa, que correspon a la persecució del cuc al jugador.

A continuació, de la **Línia 15** a la **42**, s'executen càlculs per a determinar si el cuc ha de seguir al jugador o no. Aquests càlculs determinen coses com que el jugador no estigui tan lluny que no valgui la pena perseguir-lo o que si està a una cota gaire diferent trigui més a començar a perseguir-lo (d'altra forma, si sempre que el jugador estigués lleugerament cap a la dreta del cuc, aquest es mogué cap a la dreta, el cuc acabaria movent-se inútilment i amb un resultat molt lleig).

En el cas que sí que s'hagi de perseguir l'Etho, s'executa el *method* `ChasePlayer()` a la **Línia 44**. Les seves funcions són les següents.

- **Línia 46:** Canviar la variable `isFollowing` a verdadera perquè des d'altres zones del codi es pugui saber ràpidament si el cuc està seguint el jugador.
- **Línia 47:** Reproduir l'animació de persecució, canviant la variable `condition` a 0. S'utilitza el mateix mètode per canviar les animacions que el de l'Etho, explicat anteriorment.
- **Línies 48-52:** Si el jugador està a la dreta del cuc, la velocitat és positiva (cap a la dreta) i se'l gira cap a la dreta (establint l'escala horitzontal a 1.5).
- **Línies 53-57:** Si el jugador està a l'esquerra del cuc, la velocitat és negativa i se'l gira cap a l'esquerra (establint l'escala horitzontal a -1.5)

En canvi, si ha de deixar de perseguir el jugador, s'executa `StopChasingPlayer()`, que té les següents funcions:

- **Línia 62:** Establir la variable `isFollowing` a falsa.
- **Línia 63:** Canviar l'animació a repòs (`condition == 1`).

Per detectar quan el cuc ha atrapat el jugador s'utilitza el següent codi.

```
1. void OnCollisionEnter (Collision collision) { //detectar si el cuc i Etho
   estan en col·lisió → matar-lo
2.     if (collision.gameObject.name == "Etho") {
3.         FindObjectOfType<GameManager>().GameOver ();
4.     }
5. }
```

`OnCollisionEnter()` és un *method* que detecta quan l'objecte al qual està associat aquest codi xoca amb un altre objecte (que tingui un *rigidbody*²¹), i associa aquest nou objecte a la variable `collision`. A continuació, es detecta si aquest objecte amb el qual ha xocat té el nom de "Etho" i, en cas que sigui cert, s'executa `GameOver()` a la **Línia 3**, que correspon al següent codi (que pertany a la classe `GameManager` i no la `WormMovement` com l'altre codi del cuc).

Game Over i Respawn

```
1. public void GameOver ()
2.     {
3.         if (!GameHasEnded)
4.         {
5.             GameHasEnded = true;
6.             FindObjectOfType<PauseMenu>().GameHasEnded = true;
7.             Time.timeScale = 0f;
8.             gameOverUI.SetActive(true);
9.         }
10.    }
```

Aquest codi du a terme diverses funcions:

- S'utilitza el condicional de la **Línia 3** per tal que el codi només s'executi una vegada, perquè podria ser que `GameOver()` fos cridat diverses vegades simultàniament. La primera vegada que s'executi, la variable `GameHasEnded` serà establerta com a verdadera, així que el condicional no es complirà per la pròxima vegada que s'intenti executar (`!GameHasEnded` és l'abreviatura de `GameHasEnded == false`).
- **Línia 6:** Estableix la variable `GameHasEnded` de la classe `PauseMenu` com a verdadera. La funció de `PauseMenu` és la de crear un menú de pausa quan el jugador prem la tecla d'Esc (vegeu Fig. 56). No obstant, si el jugador ha mort, hi haurà el menú de "Has mort!" (vegeu Fig. 57), així que no tindria sentit que es pogués clicar Esc per a posar en pausa el joc i que aparegués el menú de pausa a sobre del menú de mort.

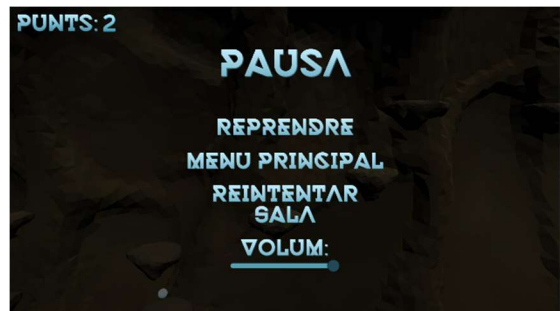


Fig. 56. Menú de pausa. Elaboració pròpia.

²¹ RigidBody: component dels motors de joc que serveix per afegir físiques i col·lisions ràpidament. Normalment en porten els personatges i objectes amb els quals s'interactua.

- **Línia 7:** Estableix el temps a 0, és a dir, para el temps. El moviment del jugador i del cuc deixen de funcionar mentre el joc estigui parat.
- **Línia 8:** Fa aparèixer el menú de mort (vegeu Fig. 57). En aquesta interfície hi ha diversos botons, cada un amb la seva funció. A continuació s'explica la més rellevant, la funció de reaparèixer o `Respawn()`:



Fig. 57. Menú de mort. Elaboració pròpia.

```

1. public void Respawn ()
2.     {
3.         Load = false;
4.         GameHasEnded = false;
5.         Time.timeScale = 1f;
6.         gameOverUI.SetActive(false);
7.         FindObjectOfType<PauseMenu>().pauseMenuUI.SetActive(false);
8.         FindObjectOfType<PauseMenu>().GameHasEnded = false;
9.         if(rP1 == null) Debug.LogError ("Respawn point null");
10.        player.position = DefRespawnPos;
11.
12.        //recarregar els estats dels objectes:
13.        if (ArrayDRPP == 2)
14.            {
15.                FindObjectOfType<WormMovement>().start = false;
16.                FindObjectOfType<WormMovement>().anim.SetInteger("condition", 1);
17.                rock2.transform.position = rock2Pos;
18.                rock2.transform.rotation = rock2Rot;
19.                rock3.transform.position = rock3Pos;
20.                rock3.transform.rotation = rock3Rot;
21.                rock4.transform.position = rock4Pos;
22.                rock4.transform.rotation = rock4Rot;
23.                worm.transform.position = wormPos1;
24.                worm.transform.rotation = wormRot1;
25.                worm.localScale = new Vector3(1.5f, 1.5f, 1.5f);
26.            }
27.
28.        if (ArrayDRPP == 3)
29.            {
30.                worm.localScale = new Vector3(1.5f, 1.5f, 1.5f);
31.                worm.transform.position = new Vector3(147.0f, -191.8f, 64.0f);
32.                worm.transform.rotation = new Quaternion(.1f, .7f, -0.1f, .7f);
33.            }
34.    }

```


En primer lloc, quan s'executa `Respawn()`, s'estableix `Load` com a falsa (**Línia 3**). Aquesta variable correspon al sistema d'emmagatzematge i posterior recuperació de dades del joc perquè, quan el jugador torni a jugar després d'haver tancat el joc, recomenci la partida des d'allà on ho va deixar l'última vegada. El sistema que guarda les dades amb seguretat a l'ordinador del jugador funciona perfectament. No obstant, no s'ha aconseguit fer funcionar el sistema de llegir aquestes dades i associar l'estat del joc de la partida actual que s'acaba d'iniciar amb el de l'anterior partida, així que, quan el jugador entri per segona vegada al joc, haurà de començar des del principi.

Línia 4: Estableix la variable `GameHasEnded` com a falsa perquè el jugador pugui tornar a morir, si s'escau.

Línia 5: Restableix el temps a 1, la velocitat normal.

Línia 6-7: Desactiva el menú de mort i de pausa.

Línia 10: Edita la posició del jugador a la posició del punt de *respawn*²² que li correspon, segons el progrés que portés en la partida.

`ArrayDRPP` és la variable que emmagatzema el número de *respawn* del jugador. Quan comença la partida, té el valor de 0 i, si el jugador mor, apareixerà de nou al principi. Quan arribi a certa posició, `ArrayDRPP` serà 1, ja que el jugador ja haurà fet cert progrés. Així doncs, quan mori reapareixerà en aquesta nova posició, no al principi del joc. I així successivament. En `WhiteCaves`, hi ha cinc punts de reparició. És important, a més a més, que no s'editi únicament la posició del jugador al reparèixer. Potser hi ha objectes amb què el jugador ha interactuat la posició i rotació dels quals també ha de ser restablerta:

- A la sala on el jugador es troba amb el cuc gegant, just abans que comenci la persecució, el jugador ha d'interactuar amb tres roques per a poder progressar. Quan el jugador aconsegueix passar la zona de les roques i es troba amb el cuc, és fàcil que mori. Si és així, la posició de les roques i del cuc, juntament amb la seva rotació, haurà de ser restablerta als valors d'inici (emmagatzemats en les variables vectorials `rock2Pos`, `rock2Rot`, `rock3Pos`, `rock3Rot`...). Això succeeix a les **Línies 13-26**.
- Quan el jugador es passa aquesta part i arriba a la següent sala (el punt de *respawn* correspon a `ArrayDRPP` 3) el jugador no interactua amb altres objectes que el cuc. Així doncs, si mor, les úniques transformacions que s'hauran d'editar són les del jugador i les del cuc (**Línies 28-33**).

²² `Respawn`: (de l'anglès, reparèixer) en els videojocs es du a terme un `respawn` quan el jugador o un personatge mor i apareix de nou a la seva posició de reparició, normalment una casa o un lloc segur.

Bolets

Tal com s'ha explicat en l'apartat corresponent al disseny del món i dels *items*, el jugador aconsegueix el poder de plantar bolets gegants per a saltar-hi a sobre i poder progressar en el joc. Aquesta és una mecànica que el jugador no té des d'un inici, sinó que l'obté cap a la meitat del joc. Per detectar quan el jugador ha d'obtenir el poder, s'utilitza el següent codi:

```
1. void OnTriggerEnter(Collider other)
2. {
3.     if (!detected)
4.     {
5.         detected = true;
6.         MushTutoUI.SetActive(true);
7.         GotPower = true;
8.     }
9. }
```

`OnTriggerEnter()` és un *method* que detecta la col·lisió entre dos objectes diferents que tinguin la propietat de produir col·lisions (és a dir, que tinguin associat un *collider*). Funciona molt semblant al *method* `OnCollisionEnter()`, explicat anteriorment). L'objecte que tingui aquest codi ha de tenir el mode Trigger activat (vegeu Fig. 58), per tal que esdevingui un objecte que pot ser

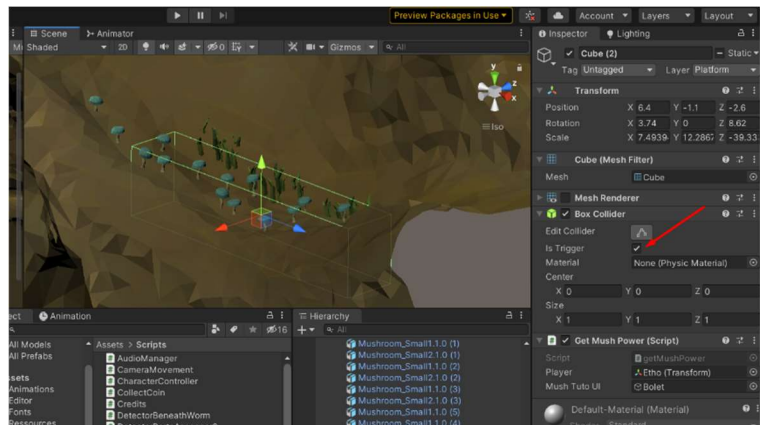


Fig. 58. **Is Trigger.** Aquest cub amb arestes verdes és transparent i traspasable, però quan un altre objecte hi entri, s'executaran les línies de codi anteriorment explicades, ja que la casella de "Is Trigger" està activada. *Elaboració pròpia.*

traspasat i que, alhora, detecti si un altre cos està dins seu. En altres paraules, `OnTriggerEnter()` és un mètode associat a un objecte, i detecta si un altre objecte entra amb col·lisió amb el primer. És molt útil per detectar quan el jugador passa per certa zona i s'hagi de produir cert esdeveniment.

En aquest cas, el codi que s'executa la primera vegada que el jugador entra dins d'aquest objecte, anomenat `DetectorGetMushPower`, fa dues funcions. En primer lloc, a la **Línia 6** fa aparèixer un element de UI, que ensenya al jugador com utilitzar el poder (vegeu Fig. 59). També posa com a vertadera la variable `GotPower` a la **Línia 7**, per ser utilitzada a continuació.



Fig. 59. Text que ensenya com plantar un bolet. *Elaboració pròpia.*

```

1. void Update()
2.     {
3.         GPower = getMushPower.GotPower;
4.
5.         if (GPower)
6.         {
7.             onPlat();
8.
9.             if (Input.GetButtonDown("Plant Mush") && PlayerOnPlat && !MushOnPlat)
10.            {
11.                StartCoroutine(StartCountdown());
12.            }
13.
14.            if (MushOnPlat && transform.localScale.x != 4.6f)
15.            {
16.                if (elapsedFrames < interpolationFramesCount)
17.                {
18.                    interpolationRatio = (elapsedFrames / interpolationFramesCount);
19.                    elapsedFrames++;
20.                    transform.localScale = Vector3.Lerp(scale1, scale2, interpolationR
atio);
21.                }
22.            }
23.        }
24.    }
25.
26. void onPlat() //see if player is on the platform so he can summon a mushroom
27.     {
28.         if (player.position.x - platform.position.x < 25 && platform.position.y - play
er.position.y < 13 && platform.position.y - player.position.y > -5)
29.         {
30.             PlayerOnPlat = true;
31.         }
32.         else
33.         {
34.             PlayerOnPlat = false;
35.         }
36.     }
37.
38. void plantMush()
39.     {
40.         MushOnPlat = true;
41.
42.         transform.localScale = new Vector3(1, 1, 1); //bolet a petita escala
43.         Vector3 destinationPos = platform.position + new Vector3(8, -
6, 0); //determinar on serà plantat
44.         transform.position = destinationPos; //plantar-lo
45.     }

```

Aquest codi està associat a cada un dels 6 bolets que es planten, que són objectes independents. Quan `GotPower` sigui vertadera (en aquest cas té el nom de `GPower` ja que es tracta d'una altra classe), el jugador tindrà l'habilitat d'utilitzar el poder de plantar bolets.

En primer lloc, `onPlat()`, *method* cridat a la **Línia 7** i executat a les **26-36**, detecta quan el jugador està a la plataforma on pot plantar el bolet. Aquesta detecció es fa a partir de distàncies (si el jugador està a certa distància de la part superior de la plataforma vol dir que no està sobre seu i, doncs, no ha de poder plantar el bolet). `PlayerOnPlat` deixa constància d'aquest estat (veritable si està a sobre la plataforma, falsa si no).

A la **Línia 9** es detecta si el jugador està prement el botó de plantar el bolet, està a la plataforma adequada i en aquesta mateixa plataforma no s'hi ha plantat ja el bolet. En cas que els tres condicionals es compleixin, es comença un altre *method* anomenat `StartCoundown()`. Aquest mètode no es troba en el codi superior ja que és una *Coroutine*, que té una sintaxi molt confusa. La seva funció és únicament esperar 0.2 segons i després fer que s'executi `plantMush()`. Si no hi hagués aquesta petita pausa, quedaria estrany que el bolet cresqués automàticament després d'haver premut el botó.

`plantMush()` (**Línies 38-45**) teletransporta el bolet a la plataforma, allà on ha estat plantat. La seva posició inicial està fora de la vista del jugador, ja que representa que el bolet no existeix fins que es planta.

Per últim s'executa el codi de les **Línies 14-22**. Tot i que pot semblar confús que prèviament s'hagi executat el codi de línies inferiors, s'ha ordenat així (amb *methods*) per fer el codi més eficient i més fàcil de llegir. La funció d'aquestes línies és fer créixer de forma suau el bolet. Inicialment, quan el bolet apareix a la plataforma, té un escalat petit, però en qüestió de menys d'un segon creix fins a arribar a la mida real.

4.2.4 Fase de proves

Aquesta etapa serveix per, a mesura que es prova el joc, corregir els errors del procés de programació i millorar la jugabilitat [43].

Generalment es classifiquen en dos tipus: les proves alpha, realitzades per un petit grup de persones que generalment estan involucrades en el desenvolupament; i les proves beta, realitzades per jugadors externs. Les primeres tenen l'objectiu de corregir defectes greus i millorar característiques fonamentals. En canvi, les segones serveixen per detectar errors menors i perfilar l'experiència de l'usuari.

En el cas de White Caves, les proves alpha les vaig dur a terme jo a mesura que s'anava desenvolupant el joc. Per poder-les realitzar més fàcilment va ser molt pràctica la funció de "Play" que Unity incorpora, perquè així es va poder anar editant el joc al mateix temps que s'hi jugava.

Les proves beta, en canvi, les van realitzar tots els jugadors que van jugar a la versió que es va publicar el 2 de novembre de 2020. Es tracta d'una versió amb algun petit error, però que no impedeix que el jugador progressi en la majoria de casos. Aquesta dura uns 30 minuts i només consta d'una tercera part de la durada total que ha de tenir el videojoc final. Va rebre molt bones crítiques de part dels jugadors, amb comentaris com "Los puzles son una pasada, el de la roca que tienes que tirar para sacar la piedra de la entrada del túnel te hace sentir como un genio", "El boss de la sanguijuela es super risas, brutal" o "La mecánica de los bolets es brutal también, que tengas que descubrir la sala medio escondida para que te den el tutorial de como se hace, creo que es genius". Naturalment, també va rebre crítiques constructives, com les següents: afegir l'opció de saltar amb la barra de l'espai perquè els salts són una mica difícils i als esquerrans els falta precisió si només es pot saltar amb la tecla W; més visibilitat en els túnels on es gateja; diferenciar més clarament l'escenari i decoracions amb les zones interectuales; afegir l'opció de córrer perquè hi ha zones on només s'ha de caminar en línia recta que es fan avorrides.

5. Conclusió i valoració personal

Afortunadament, puc afirmar que la majoria dels objectius d'aquest treball s'han completat.

En primer lloc, el procés d'aprenentatge sobre el disseny i el desenvolupament d'un videojoc, juntament amb el tirocini de la utilització dels programes necessaris, ha sigut més interessant i breu del que m'esperava inicialment. Així doncs, he tingut temps d'aprofundir encara més en aquesta matèria per millorar el meu coneixement. És clar que els nous mitjans per transmetre informació a qualsevol persona a través d'internet han facilitat notablement la meua instrucció i han permès que fos totalment gratuïta.

En segon lloc, en un principi vaig crear un petit videojoc per agafar fluïdesa amb els programes i vaig dissenyar una història que, malauradament, no respectava la mecànica jugador-personatge, així que vaig haver-la de descartar. Malgrat aquests entrebancs, vaig recomençar la creació d'un joc, per tercera vegada, i he aconseguit desenvolupar-ne una versió Beta molt treballada, amb una durada mitjana de trenta minuts i que ja acumula més de cinquanta descàrregues²³. El videojoc ja està dissenyat fins al final i ja estan creats alguns dels models 3D i animacions que no han aparegut a la primera versió. Malauradament, no he tingut temps d'acabar el joc sencer ni de millorar-lo gràcies a les crítiques que m'han fet els *beta-testers*. Malgrat que estic orgullós de com ha quedat el disseny i el joc, no finalitzaré mai White Caves perquè ja tinc pensat un nou projecte, amb un equip més gran i amb una idea més innovadora i on podré utilitzar l'experiència que he guanyat en aquest treball.

En tercer lloc, vaig realitzar una entrevista a una empresa de videojocs, on vaig aprendre sobre el procés de disseny i creació d'un videojoc, sobre la implementació de missatges en les seves obres i sobre com és ser un petit equip amb competidors com les multinacionals de l'entreteniment. Addicionalment, haver anat a l'exposició *Gameplay* del CCCB em va permetre entendre i valorar més les empreses *indies* que creen jocs; conèixer amb més detall i exemples la història dels videojocs; i el paper del gènere i els missatges en aquesta forma d'art. Així doncs, vaig intentar crear un missatge i implementar-lo passivament en White Caves. Tanmateix, no he pogut rebre crítiques sobre si està ben incorporat aquest contingut, així que desconec si ho vaig fer correctament, perquè el missatge es troba present sobretot en la darrera part del joc, que no es troba en la Beta publicada.

En quart lloc, vaig aconseguir realitzar una recerca bibliogràfica sobre els efectes psicològics dels videojocs que em va permetre conèixer amb molt més detall aquest contingut i desmitificar algunes idees que tenia malenteses. A més, la redacció de la síntesi que es troba en aquest treball em va ajudar a comprendre i

²³ White Caves està disponible gratuïtament en el següent enllaç: bit.ly/WhiteCaves

interioritzar aquest coneixement. L'oposició entre visions científiques contradictòries d'un mateix fet, com la discussió de si els videojocs provoquen o no violència, em va despertar l'esperit crític i científic per tal d'analitzar objectivament les dues perspectives i extreure'n conclusions:

- Alguns videojocs tenen molts efectes positius. Per exemple, poden ajudar a millorar les habilitats perceptives, l'atenció visual i les habilitats espacials. També poden arribar a ser una eina de formació, fins i tot d'assignatures tradicionals com àlgebra i biologia, perquè es basen en un aprenentatge multisensorial, actiu, experimental i basat en problemes. Ajuden a mitigar o fins i tot revertir la pèrdua de memòria en gent gran i, alguns, es poden utilitzar com a eina per potenciar les conductes prosocials als jugadors. Els videojocs han estat utilitzats com a forma de psicoteràpia per ajudar pacients amb càncer, teràpia ocupacional, teràpia física i en rehabilitació després d'una lesió traumàtica del cervell. Poden millorar la creativitat, però en pocs casos, ja que depèn del joc i de com es jugui.
- Molts estudis han assegurat que hi ha videojocs que comporten un augment de violència en el jugador i dessensibilització (major acceptació d'agressions físiques i empatia reduïda). D'altra banda, també hi ha nombrosos investigadors que afirmen el contrari, i asseguren que molts estudis utilitzen metodologies poc acurades i que no és clar que els videojocs augmentin l'agressivitat en la vida real. També poden produir comportaments de risc, ús de drogues i relacions pobres amb els amics i la família, i és indiscutible que existeix l'addicció als videojocs que, de fet, està classificada com a trastorn.

Referències

En aquest apartat es deixa constància de tots els documents utilitzats per a poder realitzar aquest treball, amb l'ordre en que apareixen en el text i utilitzant l'estil de citacions IEEE (Institute for Electrical and Electronics Engineers).

- [1] Facultat d'Informàtica de Barcelona, «Història dels videojocs», *Retroinformàtica*. <https://www.fib.upc.edu/retroinformatica/historia/videojocs.html>.
- [2] A. Izquierdo, «Breu història dels videojocs», *Lúdica*, 2020. <https://ludica.cat/historia-dels-videojocs/>.
- [3] Vince, «The Many Different Types of Video Games & Their Subgenres», *iD Tech*, 2018. <https://www.idtech.com/blog/different-types-of-video-game-genres>.
- [4] L. McLean i M. Griffiths, «The psychological effects of videogames on young people: A review», *Aloma Rev. Psicol. Ciènc. Educ. Espor*, p. 15, 2013.
- [5] «The psychological effects of video games.», *Medium: The Startup*. <https://medium.com/swlh/the-psychological-effects-of-video-games-d4bbdaf74783>.
- [6] N. Qiu *et al.*, «Rapid Improvement in Visual Selective Attention Related to Action Video Gaming Experience», *Front. Hum. Neurosci.*, vol. 12, p. 47, feb. 2018, doi: 10.3389/fnhum.2018.00047.
- [7] J. A. Anguera *et al.*, «Video game training enhances cognitive control in older adults», *Nature*, vol. 501, núm. 7465, p. 97-101, set. 2013, doi: 10.1038/nature12486.
- [8] D. A. Gentile *et al.*, «The Effects of Prosocial Video Games on Prosocial Behaviors: International Evidence From Correlational, Longitudinal, and Experimental Studies», *Pers. Soc. Psychol. Bull.*, vol. 35, núm. 6, p. 752-763, 2009, doi: 10.1177/0146167209333045.
- [9] M. Griffiths, «A 'components' model of addiction within a biopsychosocial framework», *J. Subst. Use*, vol. 10, núm. 4, p. 191-197, gen. 2005, doi: 10.1080/14659890500114359.
- [10] A. J. Daley, «Can Exergaming Contribute to Improving Physical Activity Levels and Health Outcomes in Children?», *Pediatrics*, vol. 124, núm. 2, p. 763, ago. 2009, doi: 10.1542/peds.2008-2357.
- [11] C. Glaubke, P. Miller, M. Parker, i E. Espejo, «Fair Play? Violence, Gender and Race in Video Games», *Child. Now*, gen. 2001.
- [12] C. Anderson, D. Gentile, i K. Buckley, «Violent Video Game Effects on Children and Adolescents: Theory, Research, and Public Policy», en *Child and Adolescent Mental Health*, vol. 13, 2008.
- [13] B. D. Bartholow, M. A. Sestir, i E. B. Davis, «Correlates and Consequences of Exposure to Video Game Violence: Hostile Personality, Empathy, and Aggressive Behavior», *Pers. Soc. Psychol. Bull.*, vol. 31, núm. 11, p. 1573-1586, nov. 2005, doi: 10.1177/0146167205277205.
- [14] N. Carnagey i C. Anderson, «The Effects of Reward and Punishment in Violent Video Games on Aggressive Affect, Cognition, and Behavior», *Psychol. Sci.*, vol. 16, p. 882-9, des. 2005, doi: 10.1111/j.1467-9280.2005.01632.x.

- [15] J. S. Lemmens, B. J. Bushman, i E. A. Konijn, «The Appeal of Violent Video Games to Lower Educated Aggressive Adolescent Boys from Two Countries», *Cyberpsychol. Behav.*, vol. 9, núm. 5, p. 638-641, oct. 2006, doi: 10.1089/cpb.2006.9.638.
- [16] Christopher J. Ferguson, «Ferguson, C.J.: Evidence for publication bias in video game violence effects literature: A meta-analytic review. Aggression and Violent Behavior 12(4), 470-482», *Aggress. Violent Behav.*, vol. 12, p. 470-482, jul. 2007, doi: 10.1016/j.avb.2007.01.001.
- [17] V. Anand, «A Study of Time Management: The Correlation between Video Game Usage and Academic Performance Markers», *Cyberpsychol. Behav.*, vol. 10, núm. 4, p. 552-559, ago. 2007, doi: 10.1089/cpb.2007.9991.
- [18] L. M. Padilla-Walker, L. J. Nelson, J. S. Carroll, i A. C. Jensen, «More Than a Just a Game: Video Game and Internet Use During Emerging Adulthood», *J. Youth Adolesc.*, vol. 39, núm. 2, p. 103-113, feb. 2010, doi: 10.1007/s10964-008-9390-8.
- [19] P. Fischer, J. Kubitzki, Stephanie Guter, i Dieter Frey, «Virtual driving and risk taking: Do racing games increase risk-taking cognitions, affect, and behaviors?», *J. Exp. Psychol.*, p. 22-31, 2007, doi: 10.1037/1076-898X.13.1.22.
- [20] C. N. de Abreu, R. G. Karam, D. S. Góes, i D. T. Spritzer, «[Internet and videogame addiction: a review]», *Rev. Bras. Psiquiatr.*, vol. 30, núm. 2, p. 156-167, juny 2008, doi: 10.1590/s1516-44462008000200014.
- [21] Ted Honderich, *The Oxford Companion to Philosophy*, 2a edició. 2005.
- [22] María Estela Raffino, «Lenguaje de Programación», 2020. <https://concepto.de/lenguaje-de-programacion/>.
- [23] Luis Olarte Gervacio, «Lenguaje de Programación», *Conogasi, Conocimiento para la vida*, 2018. <http://conogasi.org/articulos/lenguaje-de-programacion/>.
- [24] M. Maol, *Lenguajes de programación - #2 Lenguajes de bajo y alto nivel y Lenguaje Máquina*. 2018.
- [25] HTML Rules, *¿Qué es un lenguaje de programación?* 2016.
- [26] M. Maol, *Lenguajes de programación - #4 TRADUCTORES de LENGUAJES (INTÉRPRETES y COMPILADORES)*. 2018.
- [27] D. Silva, *Hablemos como máquinas, ¿Qué es el lenguaje ensamblador?* 2020.
- [28] «C# Language Specification», *ECMA international*, p. 516, 2017.
- [29] Andrew Cooke, «Computer Languages», *Andrew Cooke*. <http://www.acooke.org/comp-lang.html#sec-types>.
- [30] M. Baker, «How Do Game Engines Work?», *Interesting Engineering*, 2016. <https://interestingengineering.com/how-game-engines-work>.
- [31] A. Ahlawat, «Game Engine and History of Game Development», *Study Tonight*. <https://www.studytonight.com/3d-game-engineering-with-unity/game-engine>.
- [32] «Game development terms», *Unity.com*. <https://unity.com/how-to/beginner/game-development-terms>.

- [33] E.M. Welsh, «How to Write a Good Video Game Story», *E.M. Welsh*. <https://www.emwelsh.com/blog/write-good-video-game-story>.
- [34] Centre de Cultura Contemporània de Barcelona (CCCB), «Gameplay», gen. 2020.
- [35] «Easter Egg», *Techopedia*, 2018. <https://www.techopedia.com/definition/3786/easter-egg>.
- [36] «What is the Wim Hof Method?» <https://www.wimhofmethod.com/practice-the-method>.
- [37] E.M. Welsh, «How to Create a Video Game World», *E.M. Welsh*. <https://www.emwelsh.com/blog/video-game-world>.
- [38] E.M. Welsh, «How to Write Video Game Characters», *E.M. Welsh*. <https://www.emwelsh.com/blog/video-game-characters>.
- [39] Derakshshani, D., «Story and Narrative Development for Video Games», *Coursera*. .
- [40] «Cómo modelar 3D y formas de modelado 3D más comunes», *Infaimon*, 2018. <https://blog.infaimon.com/modelar-3d/>.
- [41] «3D modeling: an overview on various techniques», *Thepro 3d Studio*, 2018. <https://professional3dservices.com/blog/3d-modeling-techniques.html>.
- [42] «3D animation», *PC mag*. <https://www.pcmag.com/encyclopedia/term/3d-animation>.
- [43] H. Costa, «Las 7 fases más importantes en el desarrollo de juegos», *Hektor Profe*. <https://docs.hektorprofe.net/escueladevideojuegos/articulos/fases-del-desarrollo-de-videojuegos/>.

Annex 1: Entrevista a Sand Bloom Studio

Aquesta entrevista va ser realitzada per correu electrònic amb Joan Aparicio, dissenyador i gestor de comunitats en línia de Sand Bloom Studio, una petita empresa de videojocs independent amb sis treballadors de Barcelona. En els seus videojocs inclouen fortes crítiques socials i un missatge reflexiu, i és per això que em van cridar l'atenció i els vaig contactar. Addicionalment, també es va parlar amb Nomada Studio, creadors de Gris, però van denegar l'entrevista perquè amb el nou projecte no tenien temps per entrevistes.

Les seves respostes van ajudar a acabar de tenir clars certs aspectes del desenvolupament de videojocs. També em van donar confiança a l'hora de triar Unity com a programa de creació del joc i Blender per modelació i animació i no altres programes, perquè ells també els utilitzen.

-Pregunta: Què els inspira a fer crítica social en els seus jocs?

-Resposta: Creiem que els videojocs són una forma d'expressió artística, i com a tal, és capaç d'arribar a la gent d'una manera molt directe i influir molt en la seva manera de veure el món. I creiem que això ens atorga una gran responsabilitat envers aquesta gent i el missatge que els estem donant, per això una de les marques personals de Sand Bloom Studio és el fet que totes les nostres obres tenen aquest component crític envers la societat; no tant per a dir-li al jugador com ha de pensar o com s'ha de sentir, si no més aviat promovent que es qüestionin certs aspectes que d'una altra manera potser no s'hagués qüestionat.

-P: Què recomanarien a algú que es volgués iniciar en el món de fer videojocs *indies*?

-R: Molta passió, molt d'esforç i molta paciència; i que faci el màxim de xarxa possible, tant amb publisher i gent del sector, com amb altres estudis *indies*. És un món dur i ple de competidors però es guanya més donant-se suport els uns als altres que el que es perd per veure a un company de professió com a un enemic. També recomanem veure's a un mateix com a un professional des d'un inici i creure's que el seu ofici és el de desenvolupador de videojocs, tot i que sempre veurà que li queden coses per aprendre.

-P: Creuen que el seu projecte de crear una empresa independent de desenvolupament de videojocs és recomanable? Es van trobar amb moltes dificultats?

-R: Possiblement no hem escollit el camí més fàcil, però si el que creiem més convenient per a nosaltres quan vam iniciar aquesta aventura. Segurament el més recomanable és agafar experiència en una altra empresa, ja sigui una de les grans del sector o un estudi indie petit, i anar agafant experiència i fent xarxa, i després si tens inquietuds ja muntaràs el teu propi projecte personal. Però si ets un cabut com nosaltres, sempre pots intentar des d'un inici crear la teva pròpia empresa, però prepara't per sacrificar-hi hores, ja que realment és molt difícil arribar a viure d'ella sense cap referència i començant de zero.

-P: Els ha estat difícil ser coneguts i reconeguts en el món dels videojocs?

-R: Tot i que encara ens queda molt camí per arribar a considerar-nos coneguts, hem aconseguit fer molta xarxa i això a passat per assistir a tants esdeveniments com hem pogut. Des d'un inici ja vam anar a la GameLab i a la Barcelona Games World (actual Nice One Barcelona) i vam intentar reunir-nos amb el màxim de gent possible. I a la que vam poder, vam anar a la Gamescom de Colònia on vam tenir més d'una trentena de reunions de caràcter internacional. Fer xarxa és el més important i una de les fortaleses del nostre estudi actualment.

-P: Vaig veure a la seva pàgina web que hi donaven un toc d'humor. Creuen que l'humor és necessari per poder treballar en aquests temps de pandèmia?

-R: En la creació de qualsevol obra artística grupal sempre es generen tensions, és inevitable, per això és important tenir "bon rollo dins l'equip" i en això l'humor hi ajuda molt. Estem fent un producte d'entreteniment on la diversió és un factor important, almenys pel que fa a en el nostre cas; per això intentem divertir-nos al màxim fent-lo, això sí, sense deixar de prendre'ns seriosament la nostra professió. També creiem que és part de la nostra personalitat com equip, ser una mica bojos però directes amb el que fem.

-P: Quines dificultats s'han trobat al desenvolupar el seu projecte?

-R: Una de les dificultats més típiques quan crees un estudi de videojocs amb "col·legues", és el fet de treballar cadascú des de casa seva i per tant la manca de comunicació. Nosaltres des del minut zero vam llogar una oficina, però això afegia les dificultats de manteniment i pagaments de lloguers que en un inici havíem d'assumir treballant a mitja jornada i per tant no podent dedicar tant temps al projecte. També ens hem trobat amb manca d'experiència en certes parts menys tècniques en el procés de producció d'un videojoc. Un cop ja el tens fet, l'has de vendre; potser fins i tot abans d'acabar-lo. Un cop vam tenir EMMA:

Lost in Memories enllestit, no vam saber distribuir-lo bé ni fer-lo arribar al seu públic adequat. Per sort ara ens hem sabut assessorar bé de cara a futurs projectes.

-P: Venen els videojocs en suport físic? Per què?

-R: No ens ho hem plantejat mai com a estudi, ja que tampoc tenim els mitjans com per a fer-ho. Fins ara només ens hem plantejat fer jocs per a ordinador, i excepcionalment EMMA l'hem fet per a plataformes mòbils i després l'hem portat perquè també sortís a STEAM. Sempre es pot parlar amb publishers que estiguin interessats a treure el teu joc en format físic, però en tot cas ens ho plantejem quan el joc ja està acabat o força avançat en el desenvolupament, mai en l'inici d'aquest.

-P: Quins programes utilitzen en cada etapa del desenvolupament d'un videojoc?

-R: Unity per a prototipat, programació i maquetació dels nivells. Blender per a modelatge d'assets i animació de personatges, tot i que en el nostre actual projecte estem utilitzant també el MagicaVoxel, ja que per a l'estètica que li volem donar ho creiem més adequat. Per a texturitzar utilitzem el Substance painter. I SourceTree amb gitlab per al repositori i gestió de versions.

-P: Què fan si tenen diferents opinions a l'hora d'escollir una decisió important? Es posen d'acord fàcilment?

-R: A vegades ens posem d'acord amb facilitat, però quan s'allarga massa una discussió, busquem mecanismes per a resoldre el conflicte. Si no és una decisió que calgui prendre al moment intentem donar-nos temps per a reflexionar-ho individualment, o per a fer proves de les diferents possibilitats per a veure quina ens convenç més. En casos extrems o de decisió que calgués prendre si o si al moment votem i escollim la decisió més votada. Hem desenvolupat diversos sistemes de votació depenent del tipus de decisió, donant diferents punts a les opcions, podent donar valors negatius a alguna opció a les que hi estiguis realment molt en contra, etc.

-P: Quin ordre segueixen al moment de decidir el missatge que volen transmetre, les sensacions que volen que el jugador tingui, la història del videojoc en qüestió i la seva estètica?

-R: Depèn, en el cas de Sold O2ut, primer de tot teníem el missatge clar, i d'allà en va sortir el joc. Mentre que a EMMA, el missatge va acabar sortint de les primeres idees de mecàniques, i després aquest missatge va fer agafar forma a tot el joc sencer. Amb el projecte actual, DEEXIT, ha passat una mica com amb Sold O2ut, el missatge ja estava clar des d'un inici.

Annex 2: Esbossos del disseny del món

En aquest apartat s'ensenyen els dissenys del món, des de les primeres versions a mà, la versió amb dibuix digital fins al resultat final en 3D.

Dibuixos a llapis

Les següents fotografies corresponen als esbossos fets a llapis de la primera versió que es tenia pensada pel món de White Caves. La idea general s'ha mantingut intacta, però la forma de les coves i el recorregut que fa el jugador va canviar totalment. La cinquena foto correspon a l'entrada del temple, el final de la primera part. A partir d'allí, el jugador faria una ascensió pel temple fins a arribar a la superfície, però no està dibuixada aquesta part.

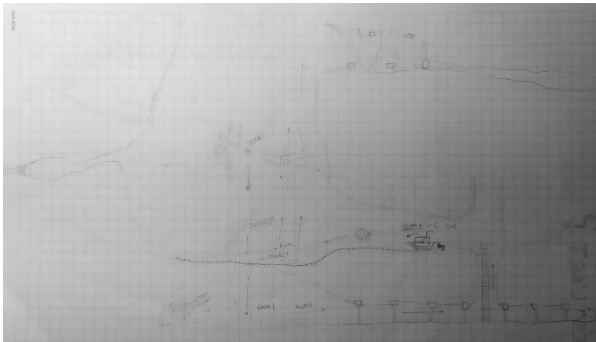


Fig. 60. Esbós del món a llapis. Primera part.
Elaboració pròpia.

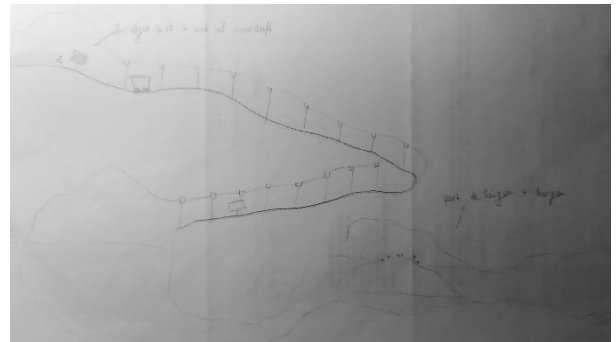


Fig. 61. Esbós del món a llapis. Segona part.
Elaboració pròpia.

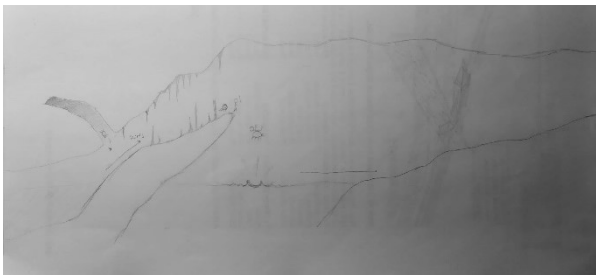


Fig. 62. Esbós del món a llapis. Tercera part.
Elaboració pròpia.

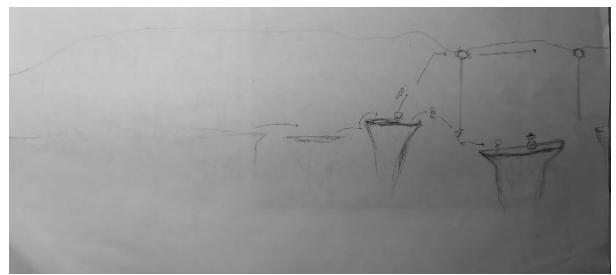


Fig. 63. Esbós del món a llapis. Quarta part.
Elaboració pròpia.

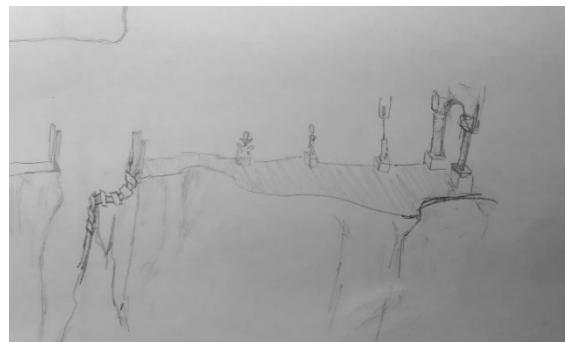


Fig. 64. Esbós del món a llapis. Cinquena part.
Elaboració pròpia.

Dibuixos digitals

Les següents imatges corresponen a la segona versió del món creada amb un *canvas* infinit. De nou, només està dissenyada la part prèvia al temple, però aquesta vegada amb més detall i més semblant a la versió final. Com es pot observar, l'escala no és consistent, i això va portar problemes a l'hora de crear el model 3D.

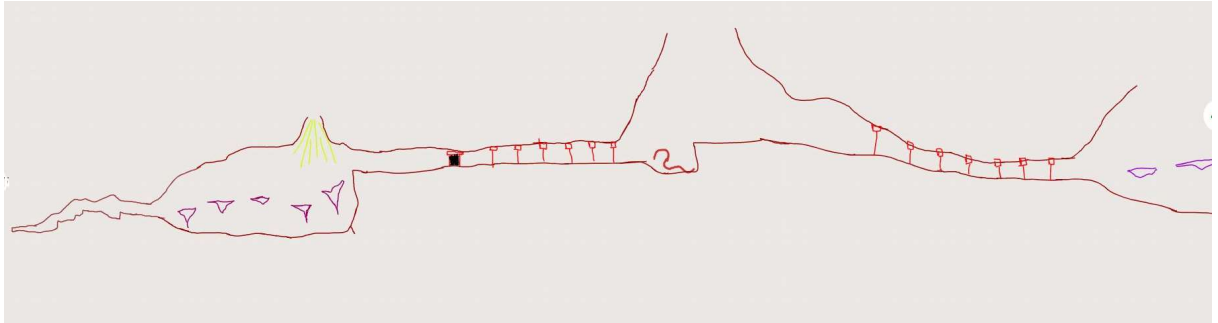


Fig. 65. Esbós del món, dibuix digital. Primera part. Elaboració pròpia.



Fig. 66. Esbós del món, dibuix digital. Segona part. Elaboració pròpia.

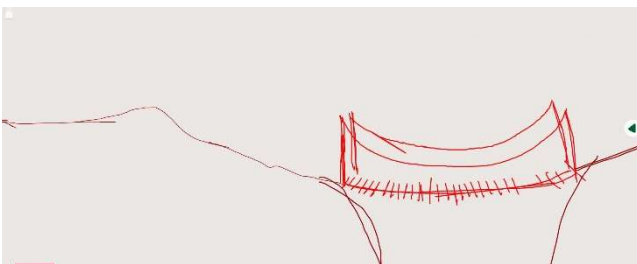


Fig. 67. Esbós del món, dibuix digital. Tercera part. Al passar, el pon es trenca i el jugador cau. Elaboració pròpia.



Fig. 68. Esbós del món, dibuix digital. Quarta part. Elaboració pròpia.



Fig. 69. Esbós del món, dibuix digital. Cinquena part. Zona dissenyada per obtenir els bolets. Elaboració pròpia.

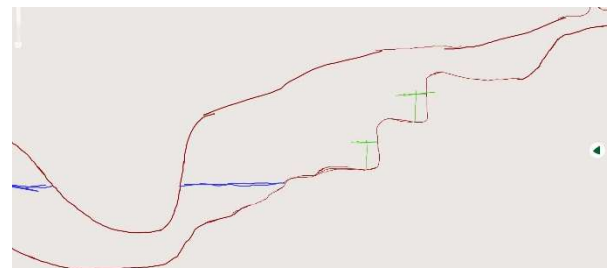


Fig. 70. Esbós del món, dibuix digital. Sisena part. Zona dissenyada per utilitzar els bolets. Elaboració pròpia.

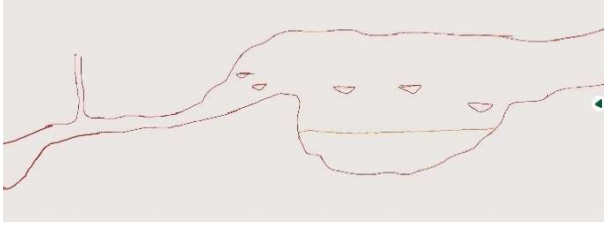


Fig. 71. Esbós del món, dibuix digital. Setena part.
Apareix l'aranya; salts entre plataformes amb lava a sota.
Elaboració pròpia.

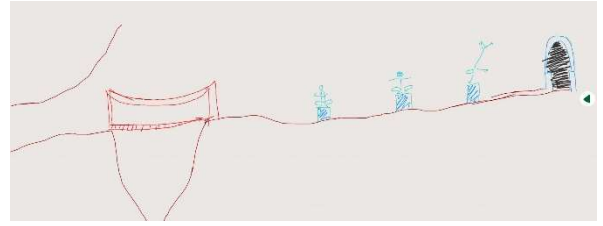


Fig. 72. Esbós del món, dibuix digital. Vuitena part.
Entrada al temple. Elaboració pròpia.

Disseny final del món

Finalment, les següents fotografies corresponen a la versió final de la fase Beta del joc, amb el model 3D del món, les textures definitives i la disposició definitiva dels altres objectes, com plataformes de pedra i fusta, rails, vagonetes, roques, caixes, elements de UI i els punts col·leccionables (esferes blau clar). L'última foto, la darrera part del món (Fig. 81), equival a la Figura 66, de l'esbós digital.

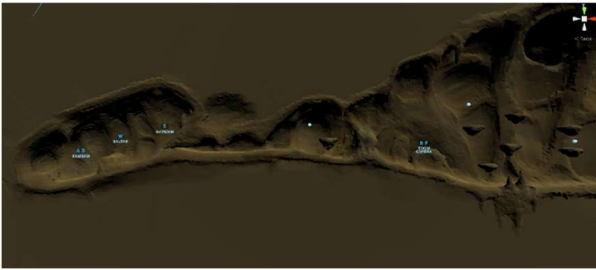


Fig. 73. Disseny final del joc. Primera part.
Elaboració pròpia.



Fig. 74. Disseny final del joc. Segona part.
Elaboració pròpia.



Fig. 75. Disseny final del joc. Tercera part.
Elaboració pròpia.

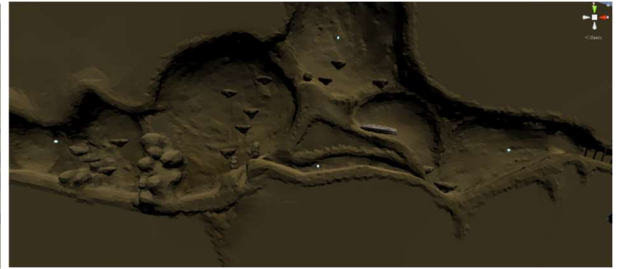


Fig. 76. Disseny final del joc. Quarta part. Puzzle amb les roques i apareix el cuc gegant.
Elaboració pròpia.



Fig. 77. Disseny final del joc. Cinquena part. Zona d'obtenció dels bolets.
Elaboració pròpia.



Fig. 78. Disseny final del joc. Sisena part.
Zona d'utilització dels bolets. Elaboració



Fig. 79. Disseny final del joc. Setena part.
Elaboració pròpia.



Fig. 80. Disseny final del joc. Vuitena part. Puzzle amb caixes de fusta per poder obtenir el col·leccionable.
Elaboració pròpia.



Fig. 81. Disseny final del joc. Novena part. A les escales on es baixa s'acaba la Beta
Elaboració pròpia.