



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Centre de la Imatge i la Tecnologia Multimèdia

# Creació d'una demo d'un videojoc en perspectiva isomètrica: Crystal Caverns

Pol Vallès Verdaguer

Director: Ismael Navales

Grau: 4t Multimèdia

Curs: 2022-23

Universitat: Centre de la  
Imatge i la Tecnologia  
Multimèdia



## Índex

Resum	5
Paraules clau	6
Enllaços	6
Índex de taules	7
Índex de figures	8
Glossari	9
1. Introducció	10
1.1 Motivació	10
1.2 Formulació del problema	10
1.3 Objectius generals del TFG	10
1.4 Objectius específics del TFG	10
1.5 Abast del projecte	11
2. Estat de l'art	12
2.1 Estudi de mercat	15
3. Gestió del projecte	21
3.1. Eines per al seguiment del projecte	21
3.1.1. Diagrama de Gantt	21
3.1.2. Kanban	22
3.2. Eines de validació	24
3.3. DAFO	24
3.4. Riscos i pla de contingències	25
3.5. Anàlisi inicial de costos	26
4. Metodologia	27
5. Desenvolupament del projecte	28
5.1 Preproducció	28
5.2 Producció	29
5.3 Postproducció	51
6. Validació del projecte	52
7. Conclusions	54
7.1 Línies de futur	54
8. Bibliografia	55
9. Annexos	59

## Resum

En aquest treball es planteja l'objectiu de crear un videojoc 3D en perspectiva isomètrica amb el motor Unity 3D.

S'avaluaran les opcions que hi ha al mercat i s'estudiaran els referents.

La metodologia usada ha estat la més extensa en el sector de la creació de videojocs, i és la de Preproducció, Producció i Postproducció.

A la preproducció es generarà el GDD del videojoc, documentació necessària per a fer un bon plantejament del joc i document imprescindible si es vol portar el joc a produir a una empresa, ja que cada persona que s'involucra a la creació d'aquest joc s'ha de llegir el GDD i ha de quedar-li clar quina feina ha de fer i com l'ha de fer.

A la producció es generaran tots els assets necessaris i es modelaran els escenaris en 3D, tot preparant-los per a la seva correcta exportació al motor de Unity. Un cop a Unity es programaran totes les mecàniques, funcionalitats i tots els puzles que conté el joc. S'usaran els scripts nadius de Unity per a controlar to i cadascun dels aspectes del joc.

Un cop tot sigui funcional s'entrarà a la part de postproducció, on es posaran els sons i es texturitzaran tots els assets i escenaris i s'il·luminarà l'escena adientment per dotar-la de bellesa audiovisual.

El resultat obtingut ha estat molt satisfactori al final de tot ja que crear un videojoc una persona sola és una tasca molt complicada i requereix de moltes hores i esforç.

El resultat és aquest, disfruteu-lo.

## Paraules clau

Videojoc, Unity, Isometric, Puzzles, Programació, Modelatge 3D.

## Enllaços

Enllaç de teaser trailer:

<https://youtu.be/n3AH74OP2sE>

Enllaç del videojoc:

<https://drive.google.com/file/d/1ihllMzljvDfw0z-pDq2uKdvX10Z08e7B/view?usp=sharing>

## Índex de taules

Taula 1. Pros i contres Unity.....	Pag. 13
Taula 2. Pros i contres Unreal Engine.....	Pag. 13
Taula 3. DAFO.....	Pag. 24
Taula 4. Riscos i pla de contingències.....	Pag. 25
Taula 5. Anàlisi de costos.....	Pag. 26

## Índex de figures

Figura 1: Perspectiva isometrica.....	Pag. 11
Figura 2: Uncharted gameplay.....	Pag. 15
Figura 3: Uncharted gameplay2.....	Pag. 15
Figura 4: Tomb raider gameplay.....	Pag. 16
Figura 5: Tomb raider gameplay2.....	Pag. 17
Figura 6: Bob Riches gameplay.....	Pag. 18
Figura 7: Bob Riches gameplay2.....	Pag. 18
Figura 8: Tunic gameplay.....	Pag. 19
Figura 9: Tunic gameplay2.....	Pag. 20
Figura 10: Càmera Unity.....	Pag. 30
Figura 11: Captura Unity.....	Pag. 31
Figura 12: Moviment Unity.....	Pag. 32
Figura 13: Model Mixamo.....	Pag. 33
Figura 14: Model Blender.....	Pag. 34
Figura 15: Cavaller Mixamo.....	Pag. 35
Figura 16: Blend tree.....	Pag. 36

## Glossari

Add-on: extensió de Blender.

Bake: Guardar la informació de relleu d'una geometria a una altra.

Build: Executable d'una aplicació generada a Unity.

Content: Contingut.

Core pilars: Pilars base d'un joc.

Features: Funcionalitats.

Feedback: Opinions, crítiques i suggeriments.

Frame: Fotograma

Game loops: Un aspecte d'un joc que es repeteix una i una altra vegada.

GDD: Document de disseny de joc.

High poly: Model 3D que conté un alt nombre de polígons.

Idle: Animació de repòs d'un personatge.

Low poly: Model 3D que conté un baix nombre de polígons.

Render: Imatge generada digitalment.

Rigging: Creació de l'estructura que permet animar un personatge 3D.

Script: Codi de programació.

Summary: Resum.

Top-down: Tipus de vista d'un videojoc en el qual la càmera està enfocant casi perpendicularment al terreny.



# 1. Introducció

## 1.1 Motivació

Els videojocs han estat una gran part de la meua vida d'es que era petit. Ja amb la Nintendo DS vaig començar a gaudir plenament de grans títols com *New Super Mario Bros.*, *Pokémon Perla...* i a partir d'aquí ha estat un no parar de jocs de moltes plataformes diferents: *Nintendo Wii*, *Playstation 3*, *Nintendo Switch*, PC i mòbil entre d'altres. Entre aquests videojocs m'agradaria destacar *Super Mario Galaxy*, és un joc que m'ha fascinat com cap altre per diferents coses. Per començar els controls i la jugabilitat és molt bona i fluïda, els puzles que s'han de resoldre són enginyosos i no són del tot fàcils, aspecte important, ja que, si fossin massa senzills, no representarien un repte i el jugador no gaudiria tant en resoldre'ls, i és un joc en el qual l'ambientació està molt cuidada, l'escenari junt amb els personatges i la música ambiental reforcen i endinsen al jugador dins el joc, creant emocions i records. Per totes aquestes raons trobo que aquest videojoc m'ha marcat un abans i un després, ja que m'agradaria aspirar en un futur a formar part d'un projecte semblant i sentir-me orgullós del resultat obtingut.

## 1.2 Formulació del problema

El que es busca a l'hora de plantejar-se aquest repte és poder aportar varietat i noves idees als jocs en perspectiva isomètrica. Es busca poder aportar bons puzles a resoldre i poder aportar noves dinàmiques a aquest tipus de jocs, ja que com no es té control sobre la càmera es poden trobar una mica limitats en aquest aspecte.

## 1.3 Objectius generals del TFG

L'objectiu a gran escala d'aquest TFG és contribuir a la indústria dels videojocs aportant un nou videojoc en el qual pugui posar a prova els coneixements apresos durant la carrera, aprendre de nous i a més a més pugui crear un projecte el que sentir-me orgullós.

## 1.4 Objectius específics del TFG

Per obtenir el producte final s'han de definir uns objectius a complir:

- Generar un GDD: Tot videojoc ha de tindre un Game Design Document on es detalli l'esquelet del videojoc, l'art, mecàniques, disseny de nivells, personatges, etc.

- Modelatge: L'entorn, els nivells, els objectes, els puzles han de modelar-se i s'ha de fer de manera que puguin ser utilitzats i programats de manera correcta.
- Texturitzat i il·luminació: El joc no només ha d'estar ben modelat i programat, sinó que una part molt important és la texturització i la il·luminació per donar encant al producte. Una bona texturització i il·luminació fa guanyar molts punts a un videojoc.
- Programació: És probablement la fase més rellevant, és el que dona sentit al videojoc, si no tan sols seria una imatge, una animació. El joc ha d'estar correctament programat i testat perquè el jugador pugui complir amb l'objectiu i no es trobi errors que li dificultin l'obtenció d'aquest.

## 1.5 Abast del projecte

L'abast d'aquest projecte va des de la part de disseny del videojoc, amb el seu respectiu document, i arriba a la producció d'una demo d'un videojoc. El producte va dirigit a tota persona que li agradin els videojocs i podrà servir com a exemple per tota persona que vulgui desenvolupar un videojoc en el futur.

En aquest projecte no es realitzarà ni el modelatge ni l'animació del personatge principal, ja que no es contempla el modelatge de personatges ni l'animació com un dels objectius. Per a obtenir els models de personatges i implementar les animacions s'usarà la plataforma Mixamo perquè compta amb una grandíssima varietat de personatges i animacions.

## 2. Estat de l'art

### Què significa que un joc tingui perspectiva isomètrica?

La perspectiva isomètrica és una tècnica utilitzada en els videojocs per representar un espai tridimensional en una vista bidimensional. Aquesta tècnica consisteix a mostrar l'escenari des d'un punt de vista obliqua, així que el jugador pot veure les diferents parts de l'entorn i les posicions relatives dels objectes i personatges. Malgrat el nom, no és necessari que la perspectiva sigui estrictament isomètrica, això vol dir que no cal que els angles entre els eixos  $x$ ,  $y$  i  $z$  formin  $120^\circ$  entre ells, de fet molts cops s'usa la perspectiva dimètrica, que implica que dos dels angles que separen els eixos  $x$ ,  $y$  i  $z$  siguin iguals.

En els jocs amb perspectiva isomètrica, els objectes més propers a la càmera són més grans que els que estan més llunyans, i això crea la sensació de profunditat i tridimensionalitat. Aquesta perspectiva es fa servir en jocs de diferents gèneres, com ara estratègia en temps real, jocs de rol, aventures gràfiques, jocs d'acció, etc.

Els jocs amb perspectiva isomètrica són populars perquè ofereixen una vista clara i completa de l'escenari, el que ajuda als jugadors a orientar-se en el joc i a planificar les seves accions. El fet de no tenir control sobre la càmera ajuda a l'usuari a centrar-se més en les mecàniques disponibles.

A més, aquesta perspectiva permet als desenvolupadors la possibilitat d'implementar detalls i efectes visuals sense comprometre massa en el rendiment del joc.

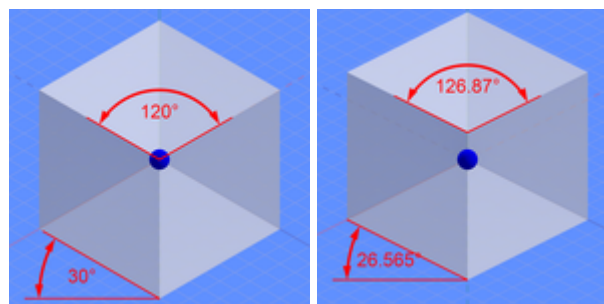
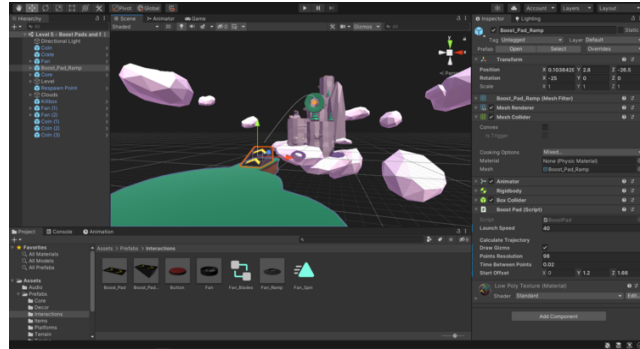


Figura 1. Perspectiva isomètrica

## Motors de videojocs

A continuació s'analitzarà els dos motors de videojocs més populars i potents del mercat.

### - Unity



Unity és un motor de videojocs multiplataforma que permet als desenvolupadors crear jocs en 2D i 3D per a una gran varietat de plataformes, com per exemple PC, consoles, dispositius mòbils, realitat virtual i altres. Aquesta eina ofereix una gran varietat de funcionalitats, com ara disseny de nivells, programació de jocs, animació, simulacions amb físiques, música i efectes de so.

El programa utilitza el llenguatge de programació C#, que és bastant conegut i fàcil d'aprendre. Es poden trobar gran quantitat de tutorials i cursos en línia. Unity també té accés a un gran llistat d'assets i continguts en línia que estan accessibles en tot moment, alguns gratuïts i altres de pagament.

Per tots els motius anteriors, Unity és una eina molt popular en la indústria, tant per petits com per grans desenvolupadors.

Alguns exemples de jocs desenvolupats amb Unity són:

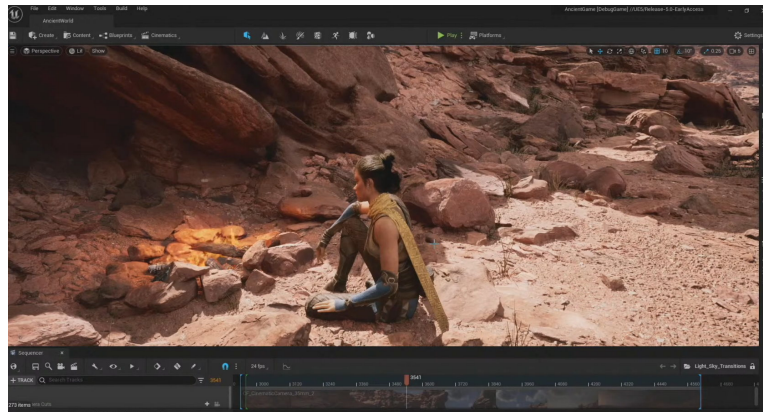
- Hollow Knight
- Rust
- Cities: Skylines
- Monument Valley

A continuació es presenta un llistat de pros i contres del motor de videojocs:

Pros	Contres
Corba d'aprenentatge ràpida	Qualitat gràfica limitada
Accés a una gran quantitat d'assets	Processament lent dels assets que afecta al rendiment del joc
Gran comunitat amb abundant quantitat de tutorials i cursos disponibles	Gran pes dels arxius que implica una càrrega més lenta

Taula 1. Pros i contres Unity

## - Unreal Engine



Unreal Engine és un motor de videojocs desenvolupat

per Epic Games, que s'utilitza per crear jocs per a una gran varietat de plataformes, com PC, consoles i altres, tant en 2D com en 3D. Aquest motor destaca per la gran qualitat visual en els jocs.

Usa dos llenguatges, C++ i Blueprint, sent aquest últim un sistema visual de programació mitjançant nodes.

Unreal Engine és utilitzat per una gran varietat de desenvolupadors, des de petits fins a grans, i és considerat com una de les millors eines de desenvolupament de videojocs del mercat.

Alguns exemples de jocs desenvolupats amb Unreal Engine són:

- Fortnite
- ARK: Survival Evolved
- Rocket League
- Dead by Daylight

A continuació es presenta un llistat de pros i contres del motor de videojocs:

Pros	Contres
Proporciona una gran qualitat visual dels jocs	És més complex d'usar que altres motors
Opció de poder programar visualment mitjançant nodes	Necessita més recursos de hardware que altres motors, afectant al rendiment
Gran flexibilitat per adaptar-se a les necessitats dels desenvolupadors, donant accés al seu codi per modificar-lo	Menys eficient carregant i processant assets

Taula 2. Pros i contres Unreal Engine

## 2.1 Estudi de mercat

### Uncharted:

Uncharted és una saga de videojocs d'acció i aventura en tercera persona que combina l'exploració i la resolució de puzles amb una història emocionant i uns gràfics excel·lents. Els jocs tracten sobre les aventures de Nathan Drake, un caçador de tresors que viatja per tot el món a la recerca d'artefactes antics i valuosos. Els jugadors han de resoldre puzles i superar obstacles per arribar als seus objectius, mentre exploren una àmplia varietat de localitzacions i escenaris diferents a través de la història.

De la saga Uncharted podem destacar la seva posada en escena, ja que la manera en què es desenvolupen les accions li dona un toc cinematogràfic al joc, a més a més, gràcies a la gran quantitat de cinemàtiques que té fan que sigui una experiència completa i immersiva per als jugadors.

M'agradaria remarcar la bona qualitat dels seus puzles, que, malgrat que en els diferents jocs alguns d'ells tinguin similituds i repeteixin mecàniques, estan molt ben dissenyats de manera que el jugador no pugui trobar la solució a l'instant, sinó que es vegi obligat a rumiar i investigar per tal de poder solucionar-los. Cal mencionar que el joc pot proporcionar algunes pistes si detecta que el jugador està encallat, cosa que evita que es frustri excessivament i perdi les ganes de continuar jugant en topar-se amb un repte que no pot solucionar.

A la interfície d'usuari es mostren clarament els controls de cada puzle per informar al jugador sobre com interactuar amb ells.

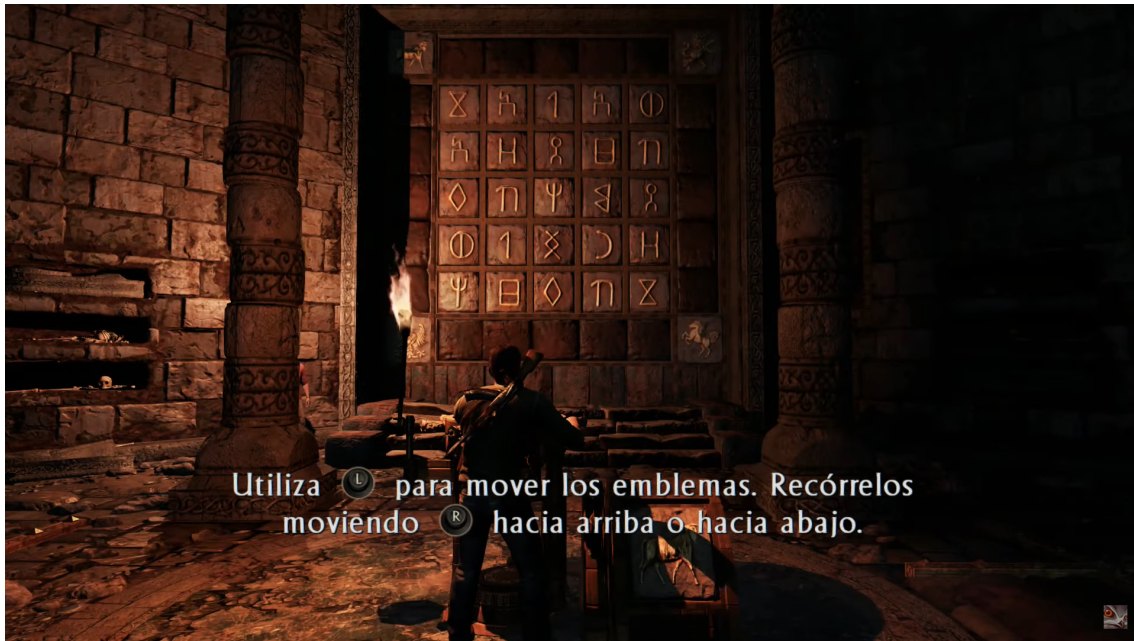


Figura 2. Uncharted gameplay



Figura 3. Uncharted gameplay2

### Shadow of the Tomb Raider:

Shadow of the Tomb Raider: Definitive Edition és un joc d'acció i aventura en tercera persona que segueix a la jove arqueòloga Lara Croft mentre investiga una illa misteriosa a la recerca d'un artefacte místic. A mesura que la trama es desenvolupa, els jugadors han de resoldre puzles i superar obstacles per arribar als seus objectius, tot això mentre es mouen per entorns perillosos. El joc destaca per la seva jugabilitat intensa i els seus bons gràfics, que presenten un món detallat i viu per als jugadors per explorar.

Els puzles són una part important del joc, aporten desafiaments i sovint requereixen que la protagonista utilitzi les seves habilitats i equipament per resoldre'ls, com per exemple, disparar una fletxa amb una corda lligada per tal de poder tensar o estirar alguna cosa.



Figura 4. Tomb Raider gameplay





Figura 5. Tomb Raider gameplay2

### **Rob Riches:**

Rob Riches és un joc d'estil arcade en el que el jugador ha de navegar per nivells plens d'obstacles, enemics i perills. Per poder superar-los ha d'aconseguir capturar totes les monedes del nivell, només així podrà avançar al següent.

Els nivells estan formats per "tiles", cosa que limita les possibilitats de moviment, però és una de les claus del joc, només poder fer moviments en les 4 direccions: endavant, enrere, dreta i esquerra. Els nivells són senzills al principi de cada món, s'introdueixen les mecàniques a poc a poc, una a una, deixant experimentar al jugador, sense dir-li res. A mesura que els nivells van avançant es combinen diverses mecàniques i és aquí quan el joc es complica de veritat.

El joc té un estil artístic estilitzat i de dibuixos animats que, sumat amb el seu gameplay el fa un joc divertit i emocionant que ofereix molts reptes.



Figura 6. Rob Riches gameplay



Figura 7. Rob Riches gameplay2

### **Tunic:**

Tunic és un videojoc indie amb perspectiva isomètrica d'aventures on el jugador pren el paper d'una guineu antropomòrfica que apareix en una platja sense saber re de re. No entén el llenguatge del lloc on està i haurà d'anar explorant el món màgic en el qual es troba, anar-se enfrontant a enemics i anar solucionant puzles per poder continuar avançant en la història.

Tunic està fortament inspirat per jocs com The Legend of Zelda, i conté la jugabilitat clàssica d'aquests títols junt amb una estètica moderna que li dona un encant extra i fa que sigui un joc per gaudir visualment.



Figura 8. Tunic gameplay



Figura 9. Tunic gameplay2

### 3. Gestió del projecte

#### 3.1. Eines per al seguiment del projecte

Per gestionar el projecte i fer el seguiment d'aquest s'usarà un diagrama de Gantt i la plataforma *Trello* com a eina per seguir la metodologia àgil Kanban, ja que són una bona combinació d'eines per tal de fer un bon seguiment del projecte.

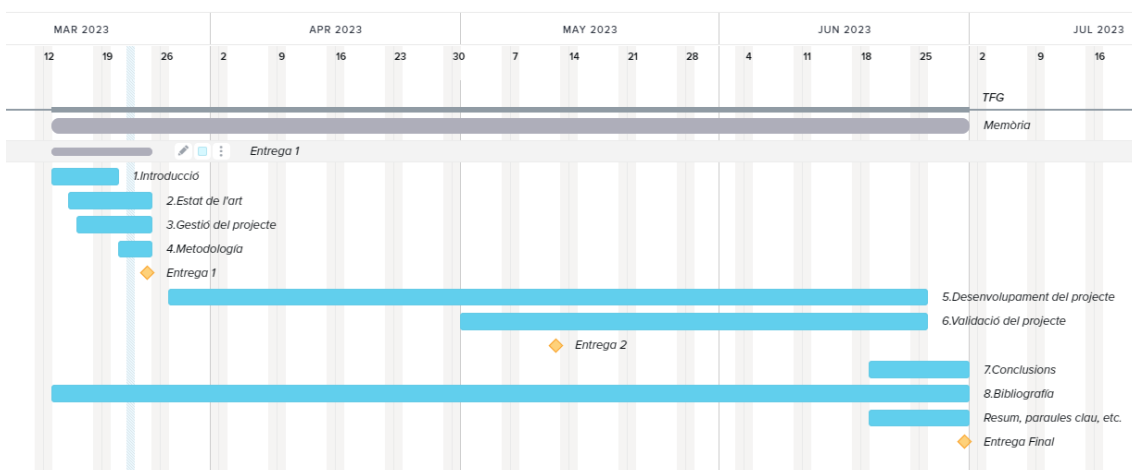
##### 3.1.1. Diagrama de Gantt

El diagrama de Gantt és una eina visual que ajuda a veure representades les tasques d'un projecte al llarg del temps, amb les seves dates d'inici i finalització, juntament amb les dependències entre les diferents taques i permet una ràpida visualització del progrés del projecte.

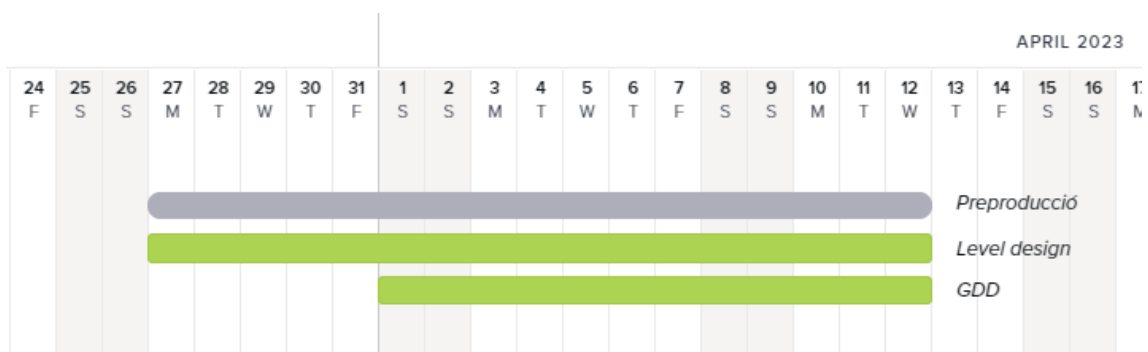
S'ha elaborat un diagrama a través de la plataforma TeamGantt que conté les tasques i fites importants pel correcte desenvolupament del projecte.

S'ha dividit en els següents grups:

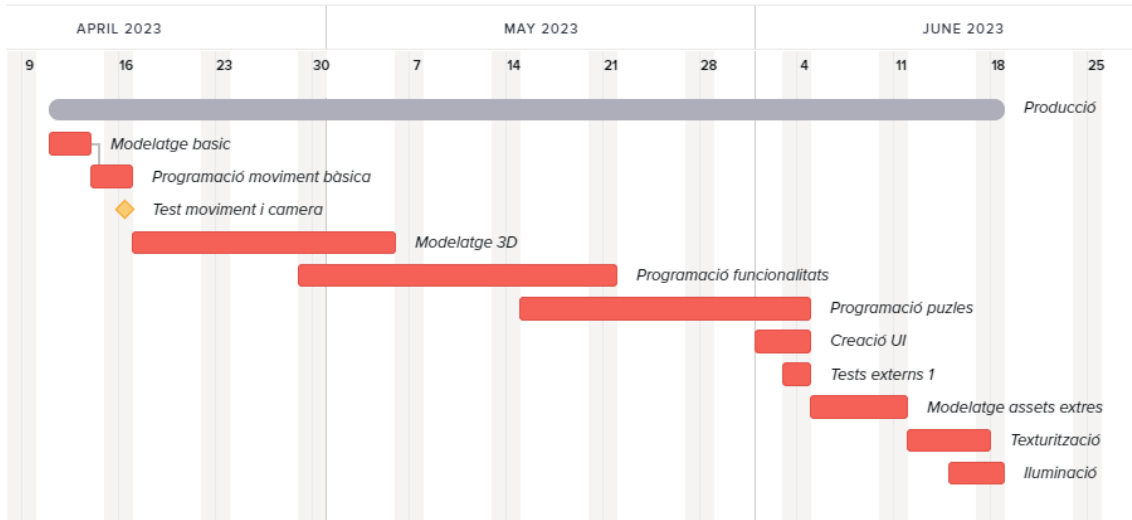
##### Memòria:



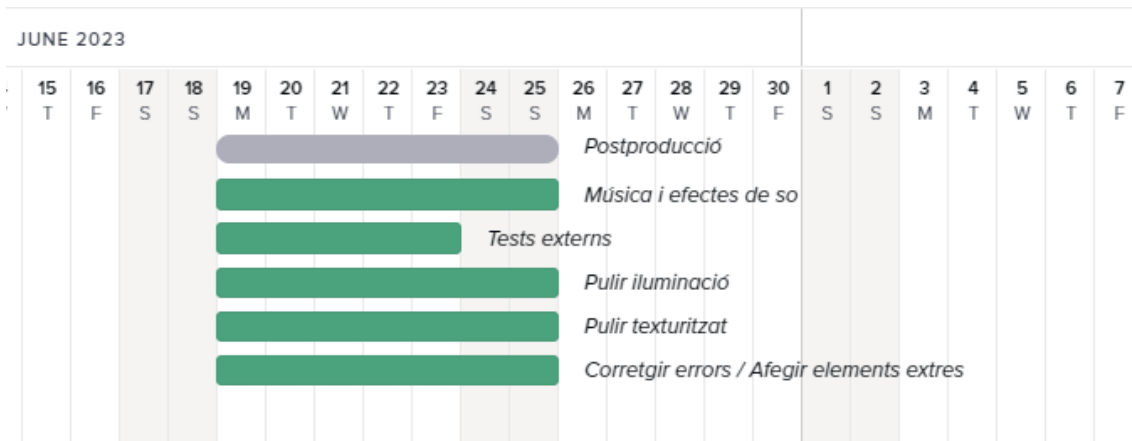
##### Preproducció:



**Producció:**



**Postproducció:**



GanttTFG\_v01.pdf

**3.1.2. Kanban**

La metodologia àgil Kanban es basa en la representació del projecte en un tauler, on cada tasca es representa com una targeta. Els taulers es divideixen en diferents columnes que mostren diferents estats, per exemple: Per fer, Fent ara, A revisar, Fet,

etc. Cada targeta es va movent de columna en columna segons l'estat en què es troba. Així es té una clara idea de les tasques que s'estan realitzant i les que venen.

Per incorporar aquesta metodologia àgil s'ha optat per utilitzar la plataforma Trello, ja que és una eina amb la qual s'ha treballat en projectes anteriors i ha funcionat molt bé per fer el seu seguiment i la planificació.

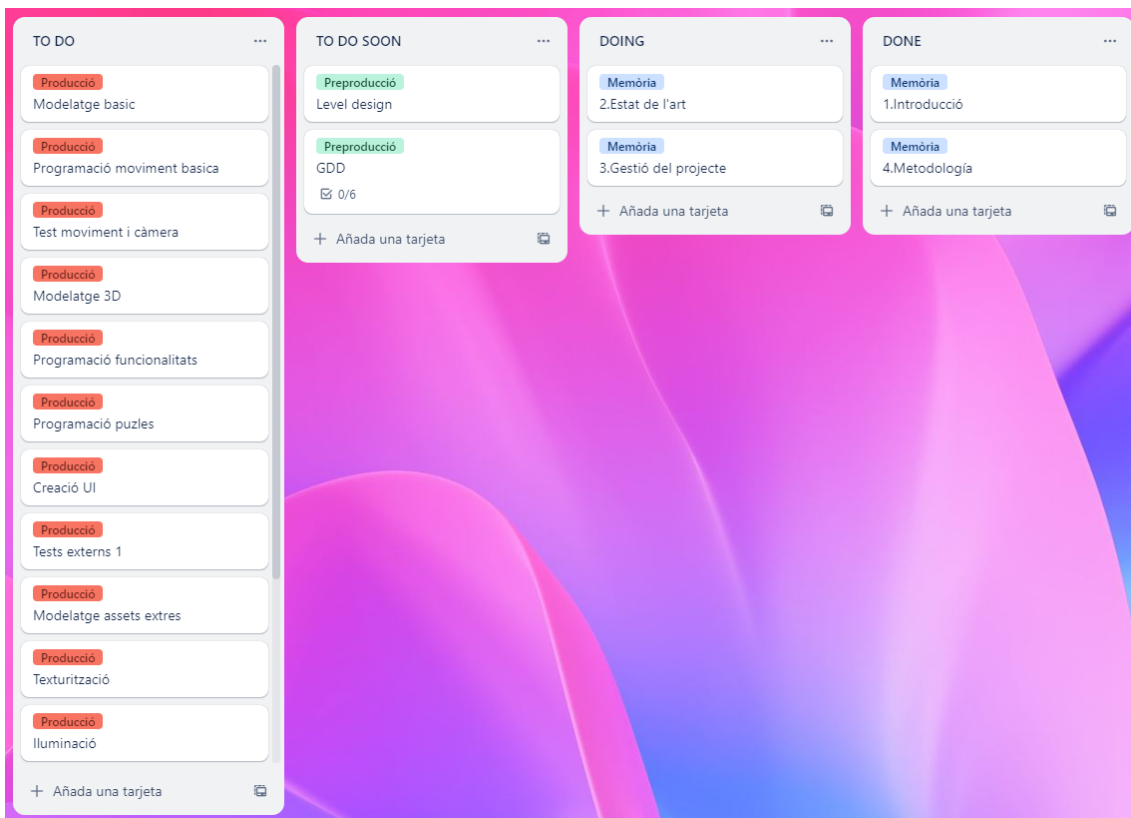
El projecte s'ha dividit en les columnes següents:

**To do:** Aquí anirà tota tasca que s'haurà de fer en un futur.

**To do soon:** Les tasques que s'han de realitzar dintre de poc temps. Serveix per tindre una visió més àmplia i preveure les tasques que s'hauran de començar dintre de poc.

**Doing:** Tota aquella tasca que s'estigui duent a terme.

**Done:** Les tasques que s'han donat per finalitzades descansaran en aquesta columna.



A part de dividir el projecte en tasques, algunes de les tasques tindran alhora diverses subtasques per tal de dividir encara més la feina i poder assolir a poc a poc els punts necessaris per completar-la.

## 3.2. Eines de validació

Per anar revisant que el projecte avança favorablement es preveuen fer diversos tests, els podem classificar en dos tipus, interns i externs.

Els tests interns es realitzaran pel desenvolupador del projecte cada vegada que es faci una modificació que impliqui la introducció d'una nova mecànica del joc. Aquests tests serviran per testejar la mecànica introduïda a fons i revisar l'aparició de possibles errors per tal de poder-ne fer un seguiment de cara a poder arreglar-los.

Els tests externs es preveuen fer amb versions més completes i estaran centrats a valorar l'experiència de l'usuari en termes com els controls del joc, la comprensió dels objectius, la comoditat, l'aspecte visual i possibles errors i suggeriments.

## 3.3. DAFO

Els punts forts i dèbils del tema i del seu desenvolupament

	Positius	Negatius
Origen Intern	<b>Fortaleses</b> <ul style="list-style-type: none"><li>- Aspecte visual atractiu i polit.</li><li>- Motivació personal.</li><li>- Organització i gestió ràpida del projecte en participar-hi una sola persona.</li></ul>	<b>Debilitats</b> <ul style="list-style-type: none"><li>- Manca de coneixements en aspectes de programació.</li><li>- Disponibilitat bastant reduïda.</li><li>- Perfeccionisme innecessari que alenteix el procés de creació.</li></ul>
Origen Extern	<b>Oportunitats</b> <ul style="list-style-type: none"><li>- Alt consum de jocs estil "stylized".</li><li>- Gran quantitat de jugadors que disposen d'un ordinador.</li></ul>	<b>Amenaces</b> <ul style="list-style-type: none"><li>- Que una empresa tregui una versió completa d'un joc semblant.</li><li>- Gran quantitat de jocs indie disponibles al mercat.</li></ul>

Taula 2. DAFO



### 3.4. Riscos i pla de contingències

A continuació es troba un llistat de possibles riscos que poden sorgir a mesura que avanci el projecte i les mesures que s'adoptaran en cas que sigui necessari.

Els següents riscos estan ordenats de menor a major importància:

Risc	Solució
Estancament continu per errors en la programació.	Buscar ajuda d'experts i companys.
La creació del personatge és més llarga de l'esperat.	Obtenció d'un model de personatge en línia.
Complexitat a l'hora d'aplicar certes mecàniques relacionades amb un puzzle.	Buscar una solució per modificar el puzzle o substituir-lo per un que requereixi mecàniques bàsiques.
Els errors de programació fan perillar la qualitat d'una mecànica en l'entrega del projecte.	Reduir el temps pensat per a desenvolupar la interfície d'usuari.
Jugadors dels tests externs són molt crítics amb la jugabilitat i no entenen l'objectiu.	Escoltar opinions, rebre feedback i suggeriments i utilitzar-los per redirigir el videojoc si escau.
El projecte no es pot completar a causa de l'escàs temps disponible.	Endarrerir l'entrega del TFG.

Taula 3. Riscos i pla de contingències

### 3.5. Anàlisi inicial de costos

A continuació tenim una anàlisi de costos partint que aquest projecte es desenvoluparà en aproximadament quatre mesos i es publicarà a Steam en finalitzar-se.

Categoria	Subgrup	Cost mensual	Cost total
<b>Programari</b>	Unity (licència estàndard)	150€	600€
	Blender	0€	0€
	Zbrush	38,99€	155,96€
	Substance Painter	19,99€	79,96€
	<b>Total programari</b>		<b>835,92€</b>
<b>Hardware</b>	Ordinador	0€	0€
	Perifèrics	0€	0€
	<b>Total hardware</b>		<b>2000€ - 3000€</b>
<b>Música i efectes</b>	Música	0€	50€
	Efectes de so	0€	50€
	<b>Total música i efectes</b>		<b>100€</b>
<b>Publicació i distribució</b>	Publicació a Steam	0€	100€
	<b>Total publicació i distribució</b>		<b>100€</b>
<b>Salari del creador</b>		1200€	4800€
	<b>Total costos laborals</b>	<b>1200€</b>	<b>4800€</b>
	<b>Cost total del projecte</b>		<b>7135,92€ - 8135,92€</b>

Taula 4. Anàlisi de costos

El cost total aproximat del projecte variarà entre 7135,92€ y 8135,92€.

## 4. Metodologia

Per desenvolupar aquest projecte s'usarà la metodologia clàssica per al desenvolupament de videojocs: preproducció, producció i postproducció.

**Preproducció:** En aquesta fase es generarà el GDD (Game Design Document) que englobarà tot el procés d'idear el joc i les seves característiques. També és la fase d'aprenentatge de les eines i tecnologies que s'utilitzaran posteriorment a la fase de producció.

**Producció:** Aquesta és la fase on es desenvoluparà tot el joc en si, es crearan els assets necessaris, el modelatge de l'entorn, els puzles, es programaran les funcionalitats del joc i es testejaran les diverses versions del projecte a mesura que es vagin implementant les mecàniques.

**Postproducció:** En entrar en aquesta fase es considerarà que el joc està complet, ja que aquí es poliran els detalls finals i es faran més tests per tal de rebre un feedback final.

## 5. Desenvolupament del projecte

### 5.1 Preproducció

En aquesta fase es van idear els diferents nivells del joc, amb els seus puzles i també es va produir el GDD corresponent.

A l'hora d'afrontar la creació dels diferents nivells, amb la forma de les coves, la distribució de les sales i dels puzles, es van fer diferents esbossos a mà, per tal de tindre més rapidesa a l'hora de realitzar canvis i ajustos, dels quals es van descartar els no desitjats i es van mantenir i millorar els desitjats. Es van anar provant diferents idees que anaven sorgint a l'hora de crear els nivells fins que al final es va triar una per tirar-la endavant a la producció.

Un cop havent realitzat els esbossos dels nivells i pensat els puzles, el joc i el seu funcionament en general, es va generar el GDD.

El GDD ha de poder ser una eina per facilitar la comprensió del funcionament i l'aspecte del joc sense haver-lo de veure, per tal era de vital importància dur-lo a terme-lo de forma correcta.

#### **Game Design Document:**

El GDD, o Document de Disseny de Joc en català, és un document que s'utilitza en la indústria dels videojocs i que serveix com a guia a l'hora de desenvolupar un videojoc. Aquest document conté tots els aspectes del joc descrits, des de les mecàniques i la jugabilitat fins a la narrativa, els gràfics i altres detalls importants.

L'objectiu del GDD és el de crear una base amb una visió clara i coherent del joc a desenvolupar per tal que pugui ser un punt de referència comú per a tots els membres que desenvoluparan el joc, com per exemple els dissenyadors, programadors, artistes i modeladors, perquè tots tinguin la mateixa visió del joc i puguin treballar a l'hora.

Normalment, els GDD es fan en anglès, ja que és la llengua base que s'utilitza en el sector i què gairebé la totalitat dels jocs comparteixen, i és la llengua en la qual parlen els diferents membres dels grans equips que desenvolupen els jocs entre si per entendre's pel fet que solen ser equips plurinacionals. Així doncs, seguint aquesta tendència de la indústria, el GDD es va redactar en anglès.

L'extensió dels GDD no és sempre la mateixa perquè depenent del tipus de joc, plataforma i temàtica, els continguts d'aquest document poden variar, ja que no hi ha cap plantilla ni ordre dels apartats establert. Normalment s'inclouen els apartats següents: Resum, Jugabilitat, Història, Disseny de nivells, Art i àudio, UX i UI i les Especificacions tècniques.

Solen tindre una extensió d'unes 30 pàgines, però com cada joc és únic i els apartats poden variar, l'extensió total també, així que no hi ha nombre mínim ni màxim de pàgines.

En aquest projecte, el GDD que es va generar era merament representatiu, per esquematitzar els objectius i les característiques del joc.

En en cas d'aquest GDD es va decidir dividir-lo en els següents apartats:


- Summary
- Core
- Features
- Content

En el *Summary* es va resumir breument el joc, explicant molt per sobre de què va, la història i l'objectiu d'aquest.

En l'apartat *Core* es van enumerar i descriure els pilars del joc. Aquest pilars són la base, l'essència del joc i no s'han de canviar durant el desenvolupament del joc o canviar però molt poc. Aquí apareixen els *core pilars*, els *game loops* i els aspectes que generen motivació al jugador.

A *Features* es van explicar les diferents mecàniques que s'utilitzen durant el joc.

Per últim, a *Content* es va afegir tot el contingut del joc, tot allò que apareix i canvia al llarg del joc, com la narrativa, els personatges, els diferents nivells, etc...

 Game Design Document\_ Crystal Caverns.pdf

## 5.2 Producció

És la fase més extensa de tot el projecte ja que abarca des de tot el modelatge, la programació del joc i els seus puzzles fins la creació de la interfície, la texturització i la il·luminació del joc.

El videojoc s'ha creat utilitzant Unity amb la versió 2022.3.2f1. S'ha utilitzat el motor de render URP (Universal Rendering Pipeline) per aquest projecte.

### **Proves de càmera i programació del moviment bàsic en un joc en perspectiva isomètrica**

De cara a tindre un primer contacte amb la creació d'un videojoc amb perspectiva isomètrica i tot el seu entorn, es va planificar a l'apartat de Gestió del projecte fer una

petita introducció amb Unity, configurar la càmera i programar el moviment del jugador.

Primer de tot es va crear l'escena en un nou projecte de Unity.

Per tal de configurar la càmera perquè el joc es vegi en perspectiva isomètrica es van haver de fer uns petits ajustos:

- En primer lloc cal rotar la càmera 45° en l'eix Y i un nombre determinat de graus en l'eix X, depenent del joc i el punt de vista que es vulgui aconseguir s'haurà de rotar més o amb menys graus. Si es vol que la càmera estigui més a prop del terra, per tant que la vista sigui més plana, el nombre de graus que s'haurà rotar la càmera en l'eix X serà menor, per exemple 15°-20°. En canvi si es vol tindre un punt de vista més picat, a l'estil *top-down*, el nombre de graus serà més elevat, per exemple 45°. Si es vol mantenir la vista isomètrica més pura, el nombre de graus a rotar la càmera en l'eix X ha de ser de 30°. En el cas d'aquest projecte la càmera es va rotar 30° ja que es va voler mantenir la vista isomètrica pura.
- L'altre cosa que és imprescindible per tal de configurar correctament la càmera és posar la projecció de la càmera en ortogràfic per tal d'aconseguir aquest efecte tant característic.
- El *Size* es pot ajustar a un valor més elevat però això dependrà de la distància de la càmera al terreny i el personatge, ja que un *size* més gran fa que a la càmera es vegi més zona. Els *Clipping Planes* també és important ajustar-los pel mateix motiu.

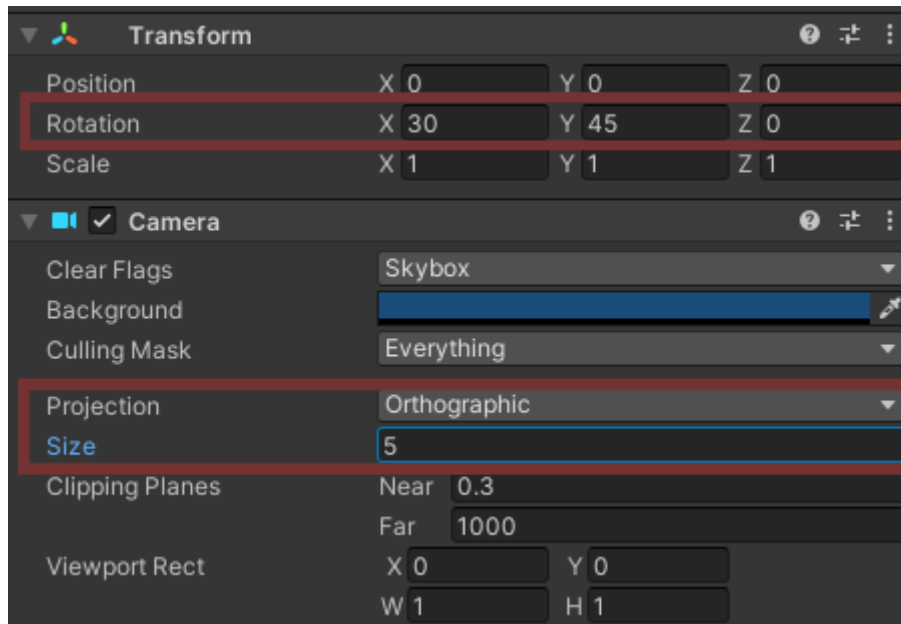


Figura 10. Càmera Unity

Un cop creada la càmera es va procedir a programar el moviment del personatge, per això es va crear un cub amb unes "ulleres" per tal de poder identificar clarament cap a quina direcció estava mirant. A part d'aquest cub es va crear un pla per on aquest cub es pogués moure.

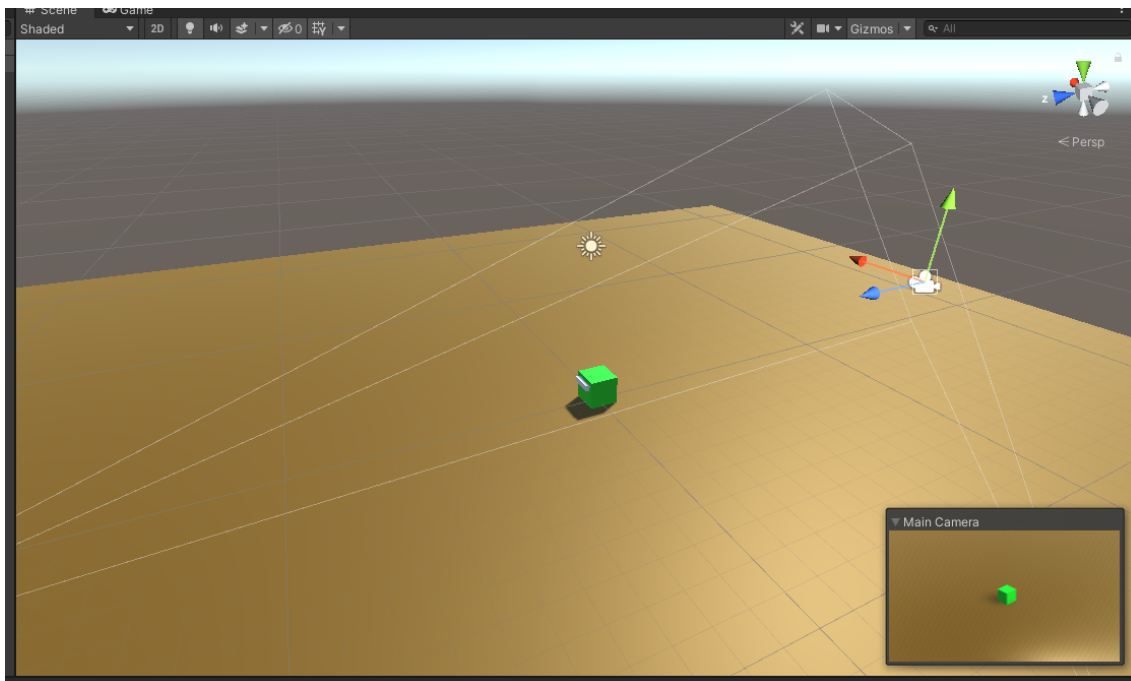


Figura 11. Captura Unity

Aquí van sorgir alguns problemes quan es va implementar la primera versió del control del personatge, i és que com la càmera estava rotada 45° en l'eix vertical, quan el jugador presionava la tecla W per moure el personatge cap a dalt de la pantalla, el personatge no es movia cap a dalt de la pantalla del jugador, sinó que ho feia en diagonal ja que seguia l'eix de coordenades de l'escena de Unity.

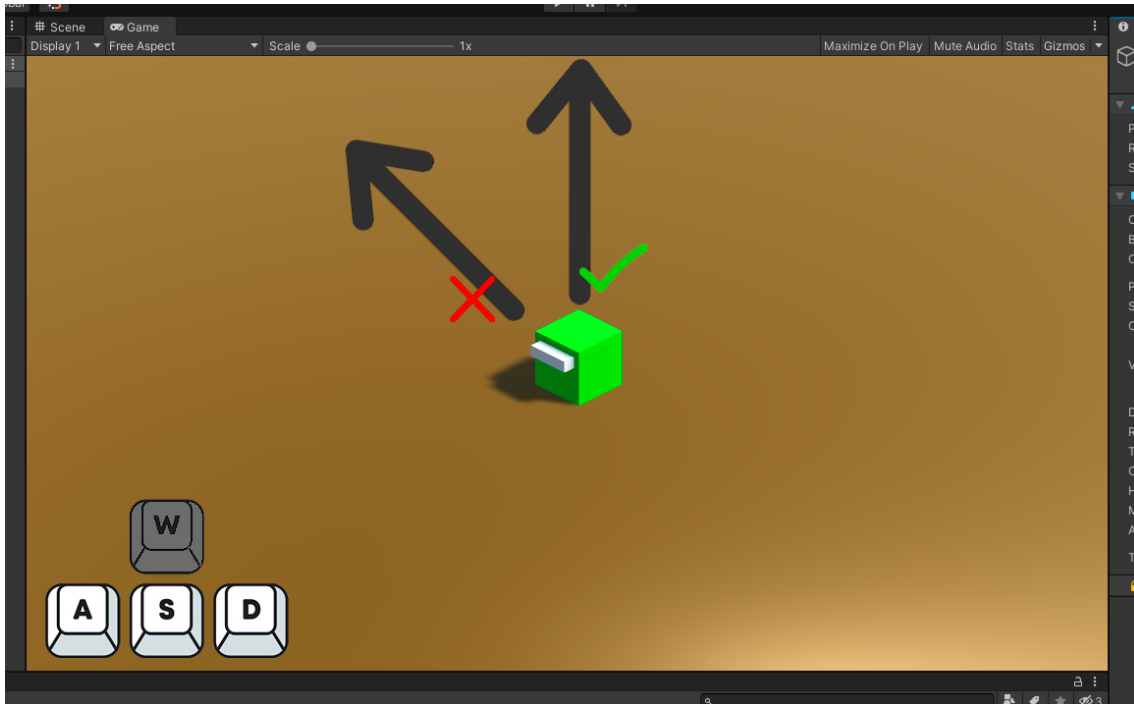


Figura 12. Moviment Unity

Per arreglar això es va haver de modificar l'*script* de control de jugador ja que el que es volia era que quan el jugador premés la tecla *W*, el personatge es mogués en vertical cap a la part superior de la pantalla. Això es va fer rotant els vectors de moviment  $45^\circ$ .

Un cop es va tindre programat el moviment bàsic del jugador el que es va haver d'aconseguir és que la càmera seguís el jugador des d'aquell mateix punt de vista. No es podia posar directament com a filla del personatge ja que si es feia així, cada vegada que aquest canviés de direcció, la càmera canviaria de direcció amb ell i rotaria, per tant sempre veuríem la càmera com des del darrere del personatge, com en un joc en tercera persona, com per exemple el *Grand Theft Auto V*. El que volíem és que la càmera es quedés quieta, que no rotés, i que només es mogués en els eixos del pla, seguint al personatge a la mateixa distància tota l'estona.

Per assolir això es va haver de fer mitjançant un *script*, al qual també se li va afegir una funcionalitat per fer el moviment de la càmera suau, ja que si no quan el jugador passava d'estar caminant a estar aturat, la càmera es parava de sobte i d'aquesta nova manera el moviment es veia molt més suau i fluït.



## Implementació del model del personatge i les seves animacions

El modelat i l'animació d'un personatge no és part dels objectius d'aquest treball, ja que és una tasca complexa i costosa en la qual intervenen molts processos, com el modelat del *high poly* i el *low poly*, el *bake* de les textures, el *rigging* i l'animació. Per això es va optar per utilitzar les eines que ens proporciona la plataforma Mixamo. Mixamo és una plataforma que ofereix una gran quantitat de models 3D de personatges que ja tenen les textures i el *rigging* fet. A part d'això també ofereix una enorme varietat d'animacions ja fetes, les quals funcionen amb qualsevol dels personatges.

Es va escollir el següent model per al personatge del jugador, junt amb les animacions idle i correr. El nom del model és *Akai E Espiritu*.



Figura 13. Model Mixamo

A aquest model se li va fer una petita modificació utilitzant el programa Blender per esborrar les fletxes que duu a l'esquena, ja que no anaven acord amb la temàtica del joc.

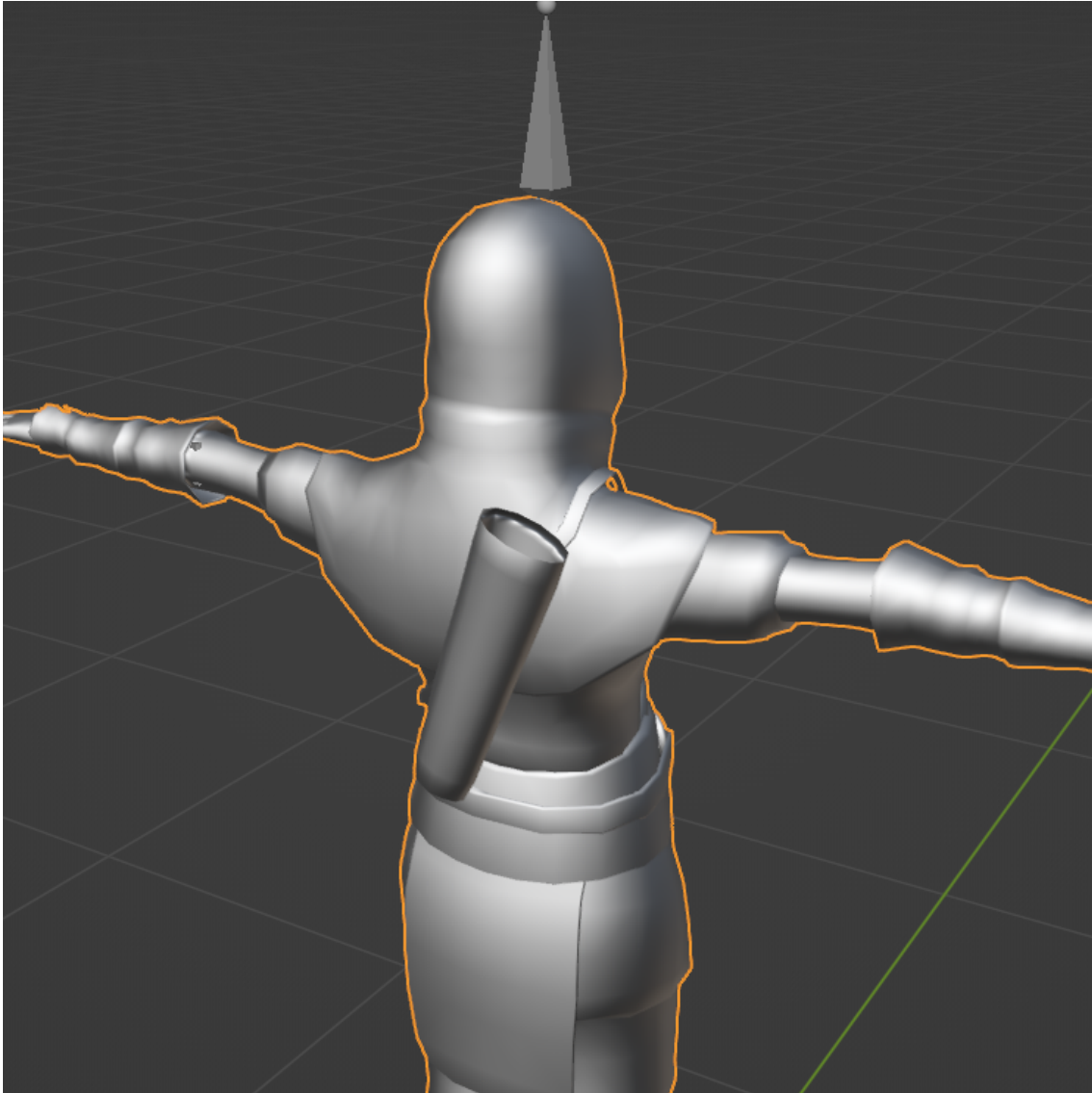


Figura 14. Model Blender

A part d'aquest model, per a confeccionar les estàtues del primer puzzle es va agafar el següent model, anomenat *Paladin W/Prop J Nordstorm*.



Figura 15. Cavaller Mixamo

A l'hora d'implementar les animacions al personatge del jugador es va fer amb un *Animator*, utilitzant un *Blend Tree* que recollia dues variables de l'*script* del controlador del personatge, i les utilitzava per seleccionar quina de les animacions s'havia de reproduir, fent les transicions corresponents entre animació i animació.

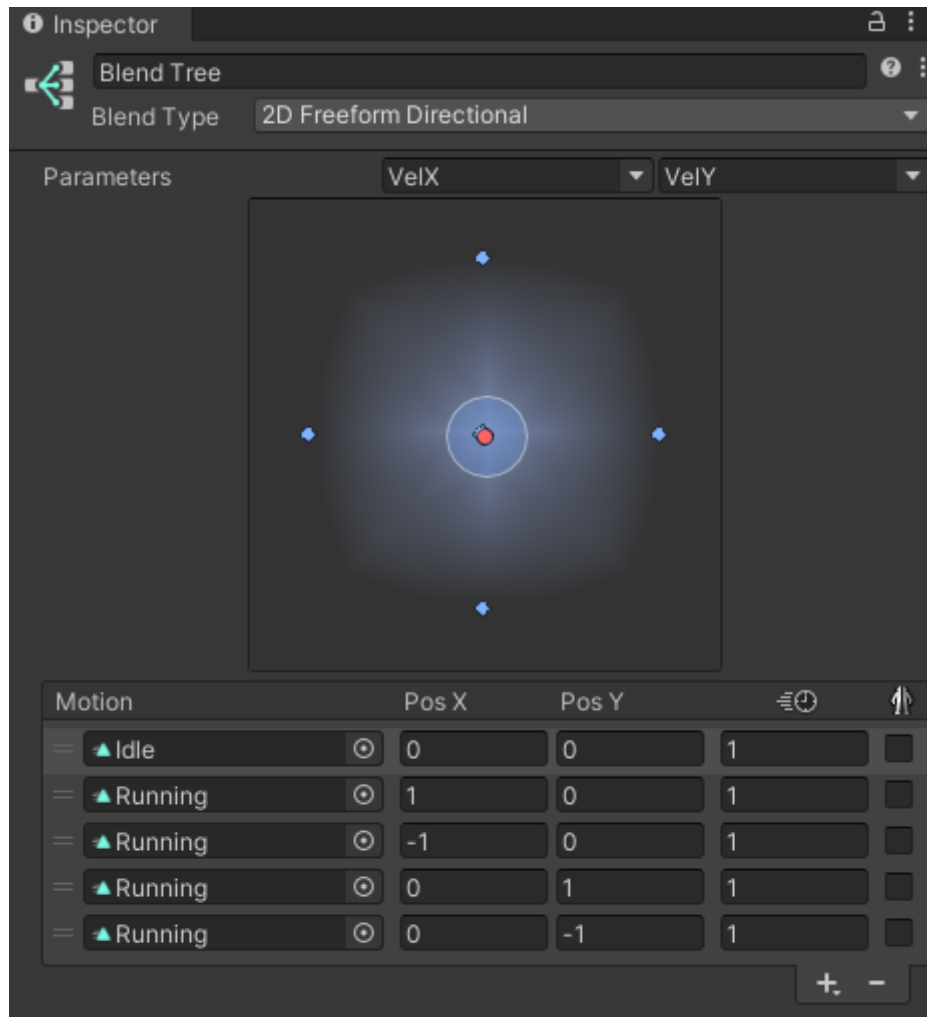


Figura 16. Blend tree

## Modelatge de l'entorn i els assets

Com a programa per modelar tota la geometria es va utilitzar Blender.

Primer de tot es va començar modelant la geometria exterior, és a dir, l'esquelet de les escenes; totes les parets, el terra i les roques per donar forma i escala als diferents nivells.

Es va modelar cada escena a un arxiu diferent, però tots tenien en comú que incorporaven el model personatge com a element unitari perquè tots tinguessin una referència de l'escala i després a l'hora d'importar les escenes a Unity sigués tot molt més fàcil i senzill.

La primera escena que es va modelar va ser la escena final, ja que era la més senzilla, doncs no incorporava cap puzzle i va servir com a una primera base per comprovar que

les escenes s'importaven bé i no hi havia cap problema. Va servir de un primer contacte per veure si les textures s'importaven correctament o si es veien molt afectades, si hi havia alguna geometria que no acceptava, etc...

Per a poder predir com s'anava a veure a Unity el model que s'estava creant a Blender, es va crear una càmera amb les mateixes propietats que la que era la càmera de Unity, una càmera de tipus Ortogràfic, la qual es va rotar 60° en l'eix X i 135° en l'eix Z, ja que els eixos a Blender estan orientats de diferent manera, i es va crear una vista separada, d'aquesta manera es podia modelar i alhora anar veient com es veuria el resultat. Aquest fet va fer que no s'hagués de modelar el que el jugador no anava a veure, ja que la càmera només tindria un punt de vista i no es podria rotar.

En la següent imatge es pot veure a l'esquerra la finestra es veu el resultat des del punt de vista que veurà el jugador, i a la dreta la finestra a partir de la qual es va modelar, on es pot veure part de la geometria que no es va modelar ja que no s'anava a veure.

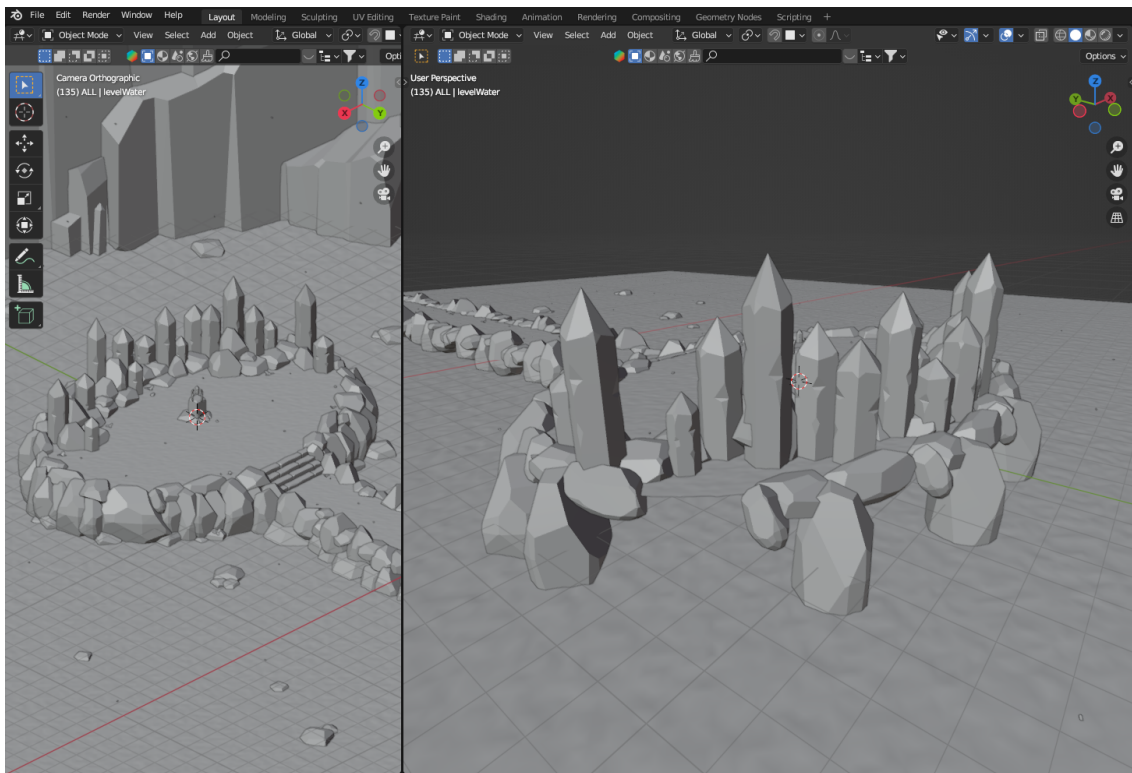


Figura 17. Escena final

En aquesta escena la geometria a modelar va ser molt senzilla ja que només es tractava de generar una varietat de roques i cristalls, els quals després es van duplicar, escalar i rotar de diferents maneres per formar diferents variacions.

La següent escena que es va modelar va ser la primera escena, on el jugador apareix per primera vegada i on es troba la gemma d'ametista.

Aquesta escena ja va plantejar un repte més gran, ja que té moltes parets i dos altures, per tant la feina afer era realment exigent.

El primer que es va generar va ser el terreny per on el jugador es podria moure, i que serviria com a base per a tota la resta. Seguidament es va dibuixar la distribució de l'escena, com si passés l'esbós del paper a la pantalla, per tal de tindre una millor referència de les distàncies i la distribució. A partir d'allà es van començar a modelar les parets, les quals es componen de dues parts, l'exterior esta composta per pedres separades i sembla que estiguin enganxades a la part interior, que es l'arrel de la paret. Aquestes pedres es van modelar amb l'ajuda d'un *Add-on* anomenat *Cell Fracture* que ve desactivat per defecte, per usar-lo només fa falta entrar a la finestra on es troben tots els *Add-ons* i activar-lo d'es d'allà. Aquest *Add-on* el que fa és agafar una geometria i trencar-la en peces, és un *Add-on* que té molts paràmetres i has de practicar una mica per saber treure-li profit del tot ja que és bastant potent. Un cop es va tenir la geometria de les parets feta es van fer els arcs per fer les portes i crear les comunicacions entre sales, el procediment va ser semblant.

Un cop situats tots els arcs al seu lloc es va procedir a generar geometria simple per aplicar una operació booleana al terreny base i a les parets, de cara a crear el desnivell i fer forats als murs perquè el jugador pugui passar d'una sala a l'altre.

Tot seguit es va crear l'altar, una geometria simple a la que després se li van fer imperfeccions i un aspecte desgastat, com a tota la resta d'assets.

Un cop aplicades les operacions booleanes es van acabar d'omplir els espais que s'havien fet per posar una mica de decoració, com per exemple el mini llac subterrani, que incorpora unes plantes i roques de l'anterior escena.

L'últim que es va incorporar eren les estàtues. Ja s'havia elegit i importat el model del cavaller però faltava crear-li dues armes, ja que el model incorporava un escut i una espasa, però faltava la maça i la destal. Aquestes dues armes van ser creades prenent com a referència la geometria de l'espasa.

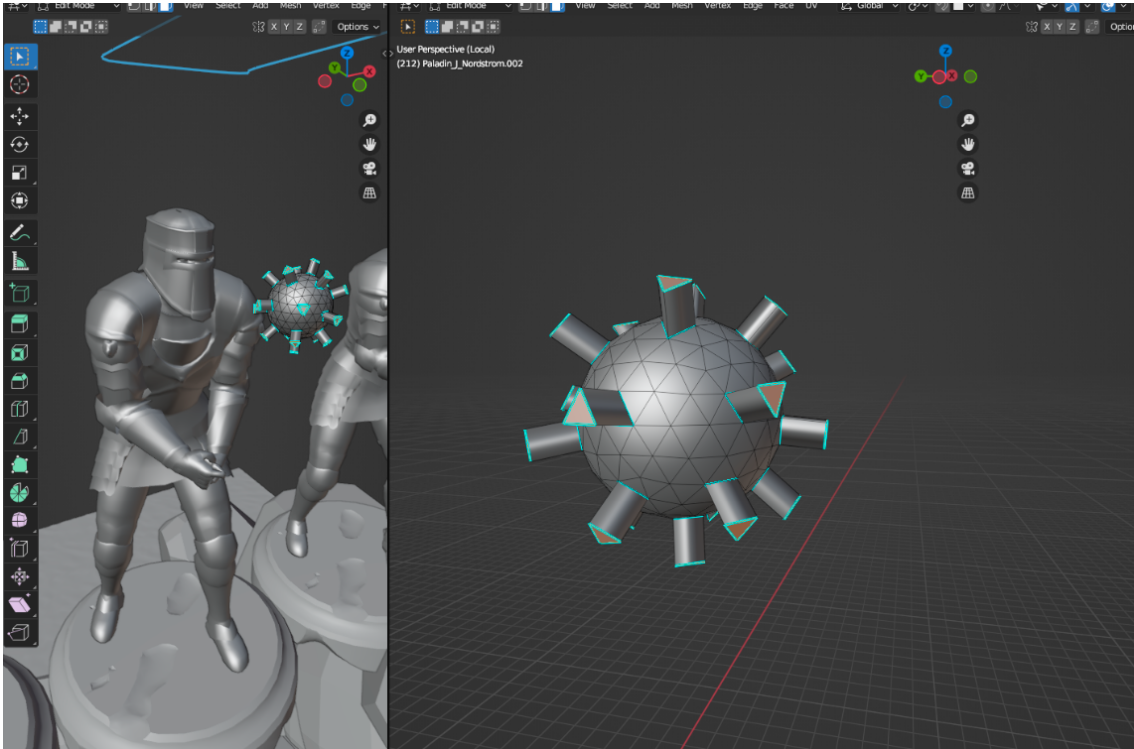


Figura 18. Modelatge d'armes

Per crear la pista que et dona la solució del puzzle de les armadures es va, fer una sèrie de renders de les armadures i les armes que posteriorment es van compondre per crear la textura final.

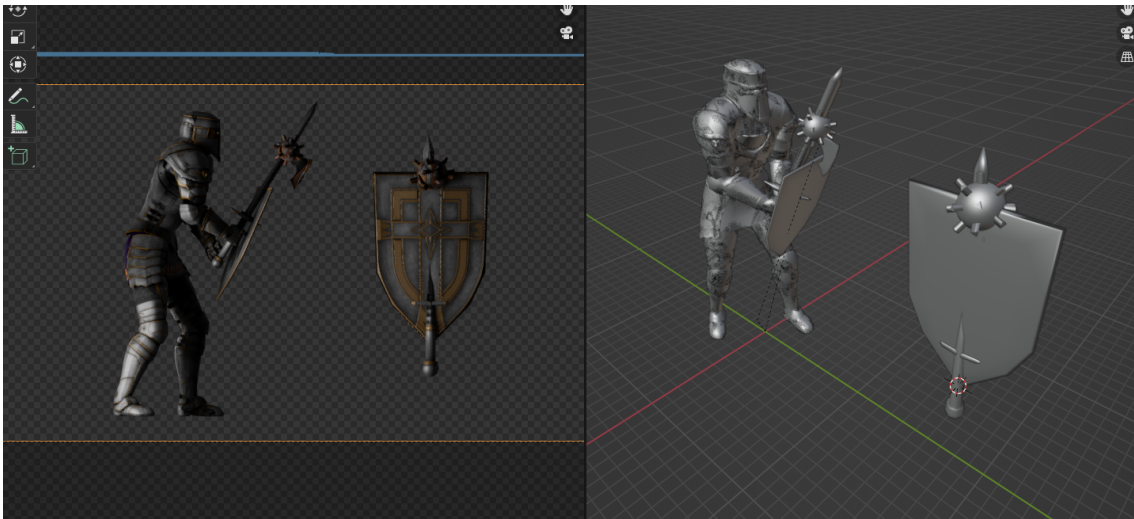


Figura 19. Escena pista

Aquí tindriem el resultat final de l'escena un cop modelada tota la geometria.

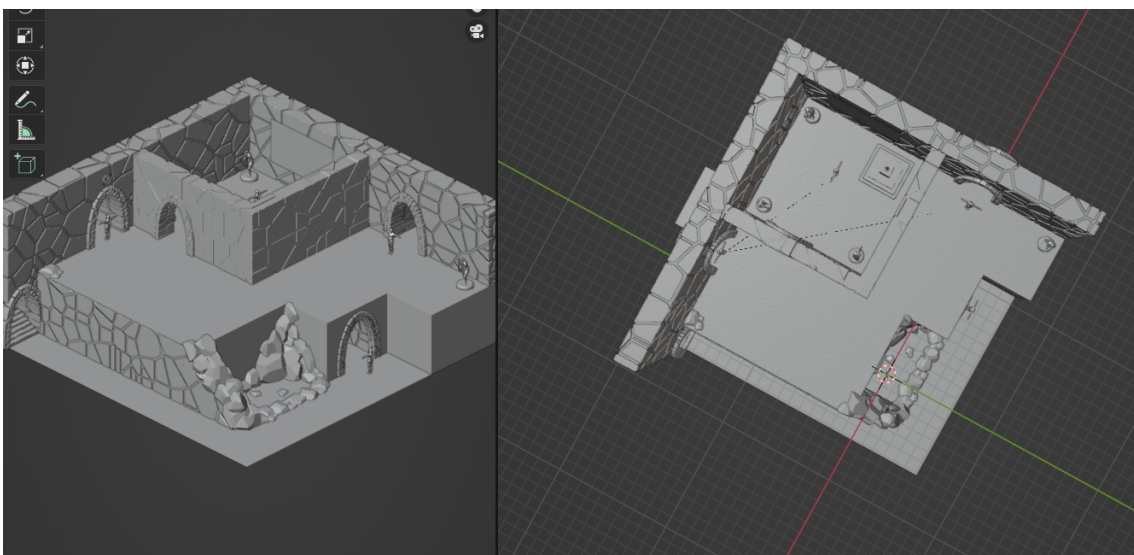


Figura 20. Resultat escena1

Seguidament es va crear l'escena que se situa a sota de la que s'acabava de modelar.



Aquesta escena, al ser plana i sense desnivells, no va costar tant de modelar com l'anterior. Per generar les parets i els arcs es va utilitzar el mateix mètode ja comentat.

Per a generar les plaques es va crear una primerament, i se li va aplicar una operació booleana per crear el relleu del dibuix que es veu, aquest dibuix representa la forma del panell que es troba a la porta final i que el jugador ha de completar col·locant les tres gemes per poder activar-ho i poder arribar a l'escena final.

Un cop creat el relleu es va procedir a crear el desgast a la placa, per embellir-la.

Un cop acabada de modelar se li va aplicar una operació de matriu per generar tota la graella de plaques i es van esborrar les que sobraven. Cal remarcar que se li van haver d'aplicar diferents textures a diferents cares, i tot això es va haver de fer a Blender ja que Unity no permet assignar un material a certes cares determinades, l'objecte ha de vindre ja amb els diferents materials assignats a cada cara i després ja et deixa modificar aquests materials.

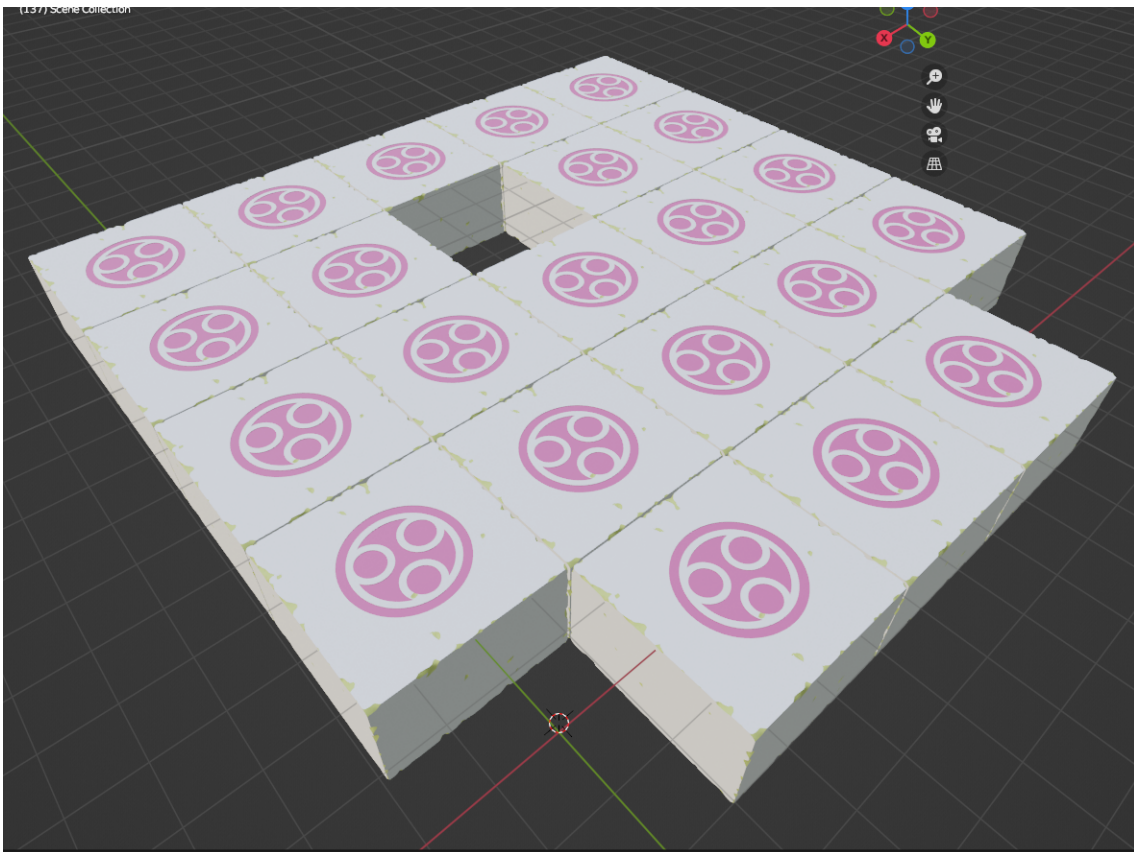


Figura 21. Plaques

Un cop tot modelat aquest va ser el resultat.

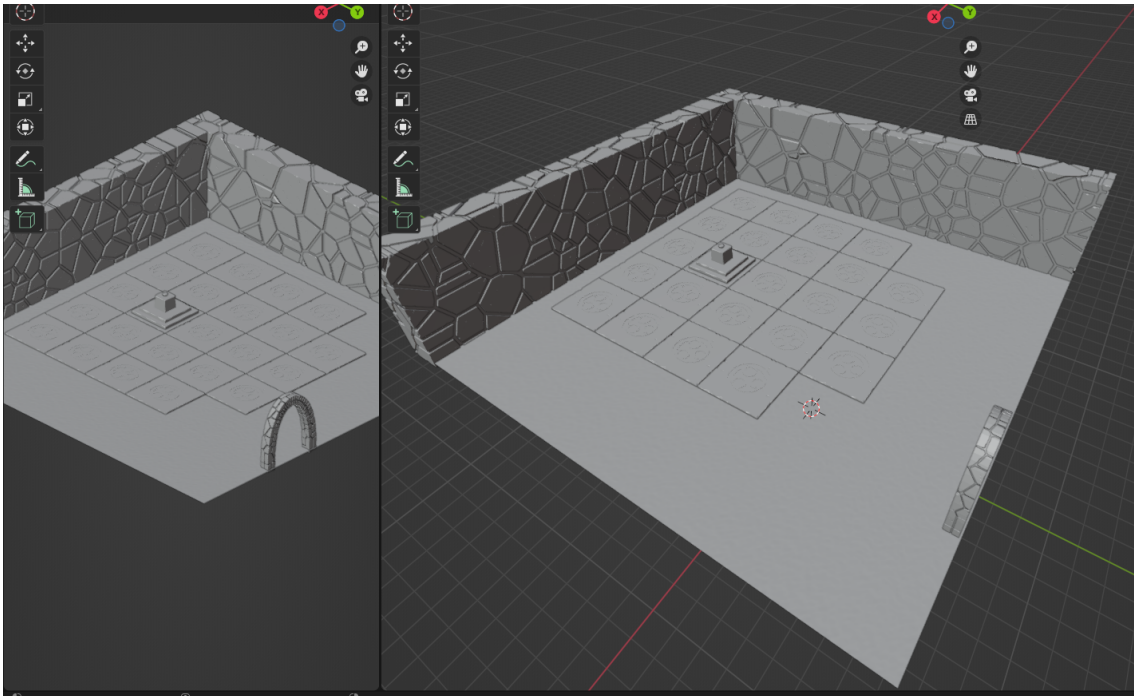


Figura 22. Escena plaques

La última escena a modelar va ser la que contenia la porta final.

Pel modelatge de les parets i dels arcs es va seguir amb el mateix mètode, i per la creació de l'escala que porta al jugador cap al nivell inferior de la primera sala es van fer servir els assets de l'escala que es va haver de crear a l'inici, quan es va realitzar el modelatge de la sala final, ja que aquesta incorpora una petita escala de tres graons. Aquesta escala es va utilitzar a la primera escena per donar-li més sentit i unió a la transició entre les dues escenes.

Per crear la porta es va fer a partir d'un cub al qual amb l'eina del ganivet, se li van fer els talls per donar-li aquest efecte de fusta. Seguidament es va crear el panell on aniran a parar les gemmes. Es van utilitzar operacions booleanes entre diversos cercles per crear la geometria. Aquesta geometria es va utilitzar en les tres escenes prèvies a l'escena final per decorar i mostrar al jugador quin és l'objectiu final.

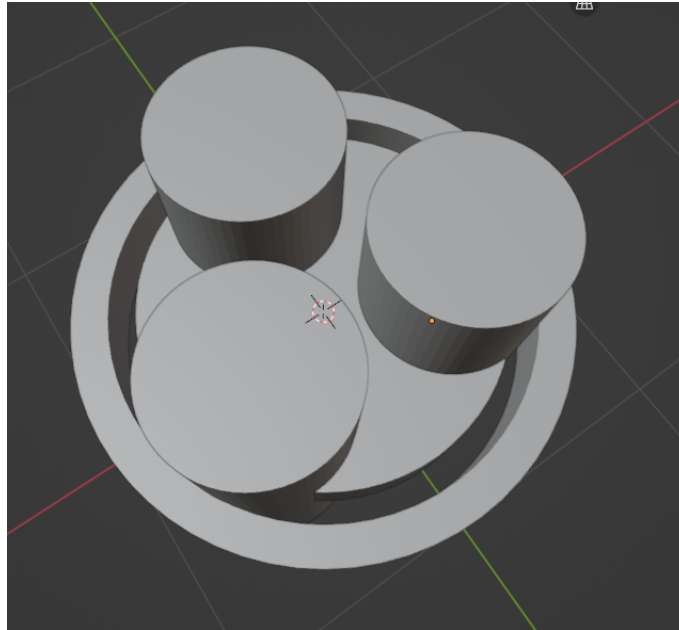


Figura 16. Blend tree

A l'hora de modelar el puzle dels cubs, es va generar un cub, que es va subdividir i es va arrodonir una mica, en aquest se li van sumar els gravats descrits a sobre a cada una de les cares. Seguidament es va crear la base i la pedra que tenen a sobre, per donar-li més sentit i cohesió, ja que sinó sembla una pedra que està flotant allà al mig sense sentit.

Un cop creades les imperfeccions es duplica fent servir el modificador de matriu per crear les altres 4 còpies que completen el puzle.



Figura 16. Blend tree

Un cop modelats tots els assets de l'escena, aquest va ser el resultat.

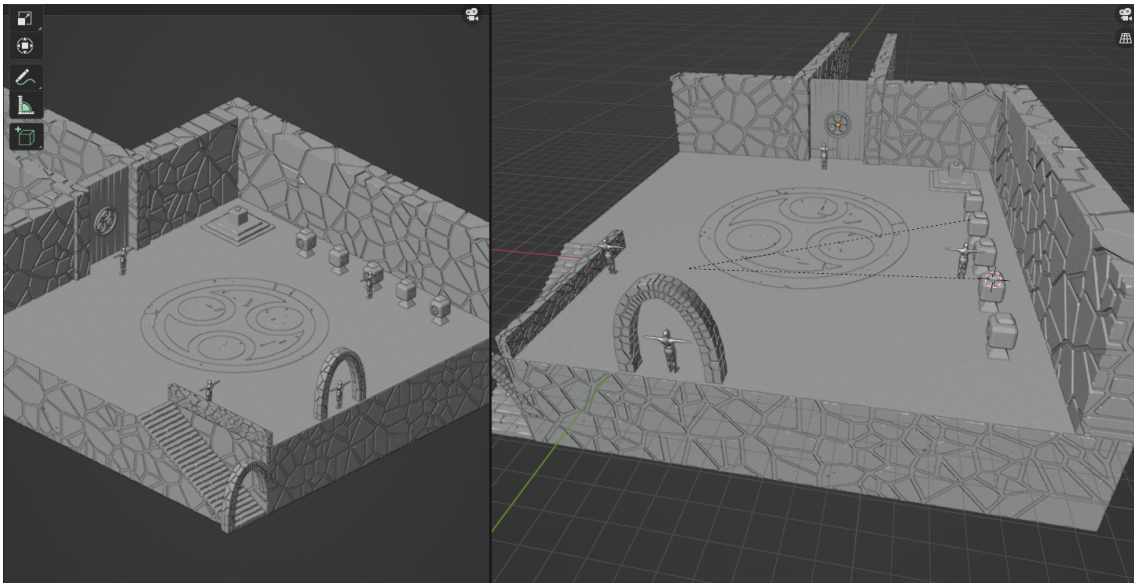


Figura 16. Blend tree

El que quedava era modelar les gemmes i el cristall final. Per fer-ho es van consultar videos sobre el modelatge d'aquestes joies ja que al ser un element tan important del joc, es volia que siguessin el mes fidels a la realitat, també per donar una sensació de pulidesa a la hora de crear el joc.

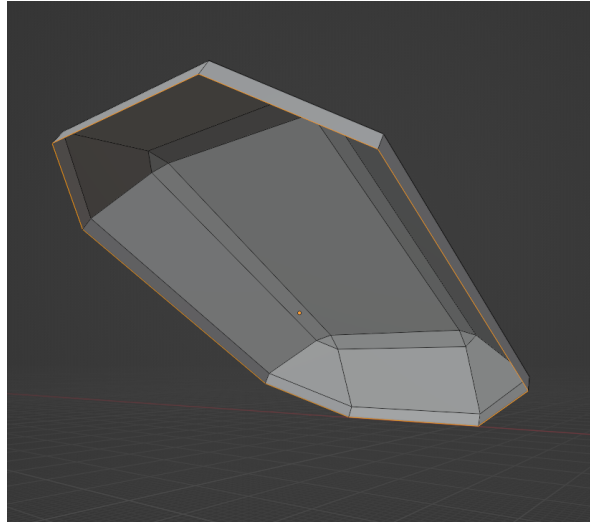


Figura 16. Blend tree

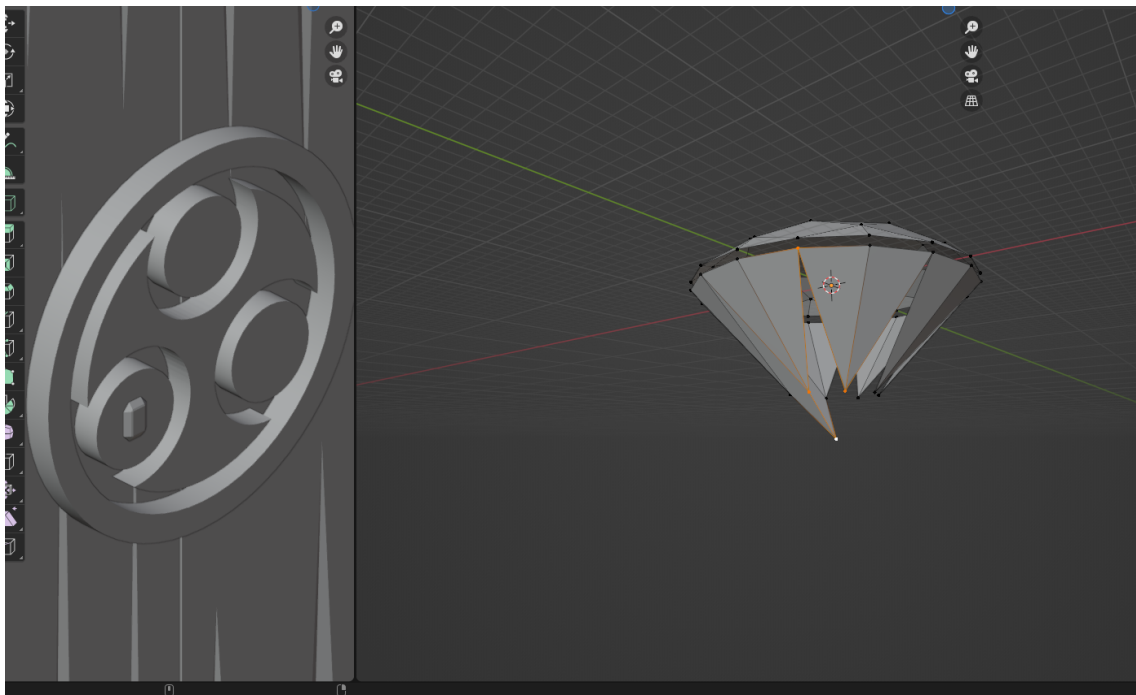


Figura 16. Blend tree

Tot seguit es pot contemplar el resultat final.

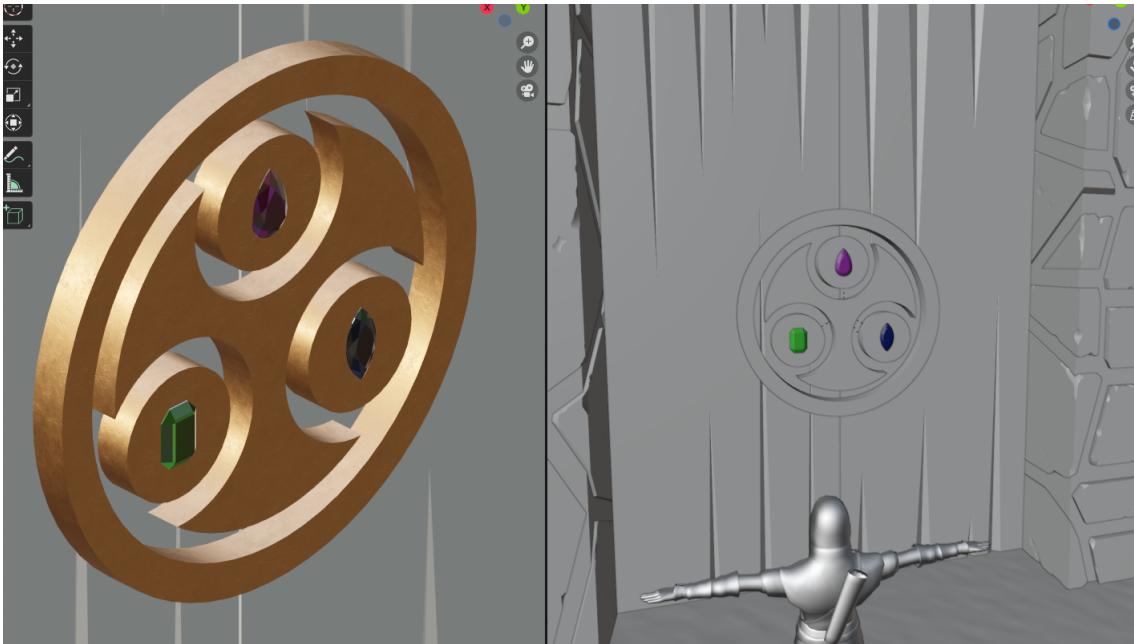


Figura 16. Blend tree

### Programació de les funcionalitats i els puzles

Aquí es farà un repàs de la programació de les funcionalitats i dels puzles del videojoc.

La primera funcionalitat que es va implementar va ser la d'ocultar la paret de la primera escena quan el jugador entrés a la sala. Es va provar de fer de moltes maneres i utilitzant molts recursos diferents, amb shaders, utilitzant raycast, etc... Finalment es va optar per una opció més senzilla, duplicar la geometria, per tant hi havia una paret que tenia els materials opacs i l'altre versió tenia els materials transparents i utilitzar un *box collider* com a *trigger* per tal de fer que quan el jugador entri al *box collider*, fer que la geometria opaca s'amagui i es mostri només la transparent, i quan el personatge surt del *collider*, torna a l'estat inicial.

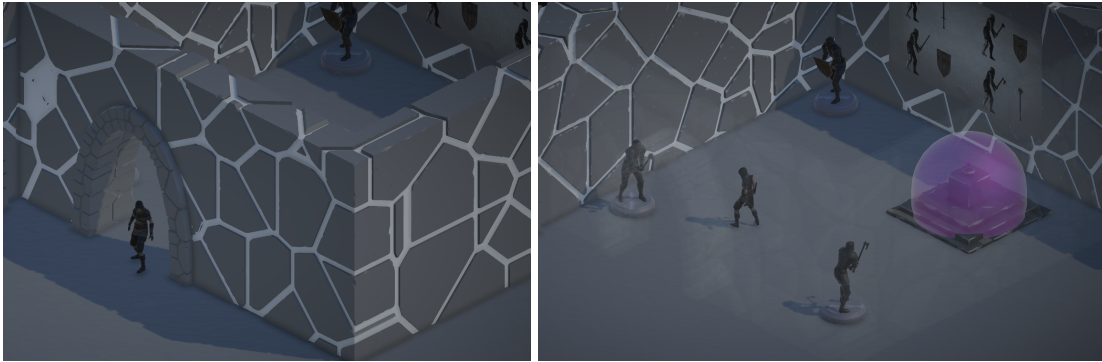


Figura 16. Blend tree

El primer puzzle, el de les armadures, el seu funcionament és més o menys senzill. El puzzle està fortament inspirat en un dels puzzles del videojoc *Uncharted 3: La traició de Drake*, enllaço el vídeo on es pot veure el puzzle [aquí](#).

Cada armadura conté un *box collider* amb *trigger* activat i 4 posicions predefinides, corresponents als 4 eixos, de les quals una sola posició és correcta. Quan el jugador entra al *collider* i prem la tecla E, l'estàtua rota amb un *Slerp*, ja que fa un efecte de desacceleració quan acaba de rotar i el *Slerp* ens permet fer això.

Per a cada rotació el codi fa una comprovació de totes les estàtues si la seva posició actual equival a la posició correcta, quan aquesta condició es compleix per totes les estàtues a la vegada, la cúpula que protegeix la gemma es desactiva i el jugador pot agafar-la.



Figura 16. Blend tree

El segon puzle, el dels cubs que roten, és bastant més complex. Aquest puzle està inspirat per un puzle del videojoc Genshin Impact, enllaço el vídeo on es pot veure el puzle [aquí](#).

Aquí el que es genera és una matriu que conté tots els cubs, per tant cada cub té assignat un índex, del 0 al 4.

El que succeeix quan el jugador s'apropa a un d'ells i prem la tecla E, és que el codi identifica a quin dels 5 cubs s'ha dut a terme la acció, i depenent de la resposta es roten uns cubs o uns altres. Per exemple, si el jugador s'apropa al cub amb índex 0 i dona l'ordre de rotar, el codi reb que l'ordre s'ha executat al cub 0 i executa les ordres de rotar els cubs amb índex 0 i 2. Cada cub rota una sèrie de cubs diferents, és per això que la gràcia d'aquest puzle és analitzar els patrons de rotació de cada cub per poder mirar d'orientar-los tots igual, si el jugador intenta solucionar per atzar aquest puzle, trigarà bastant.

Aquesta rotació es fa mitjançant una *Coroutine*, que vé a ser un mètode semblant al *Slerp* que s'ha vist abans i el que es fa perquè no es superposin és utilitzar una bandera, que és una condició, per tant si aquesta es verdadera significa que els cubs estan rotant i no es pot executar una altra acció fins que aquests deixin de rotar.

Per comprovar si els cubs estan orientats tots igual, el que es fa és comprovar per cada rotació, si el valor de rotació de tots els cubs és el mateix, és a dir, si tots els cubs tenen



el valor de rotació de 90 graus, per exemple, el puzzle es dona per solucionat i, com al puzzle anterior, la càpsula que guarda la gemma desapareix perquè el jugador la pugui agafar.

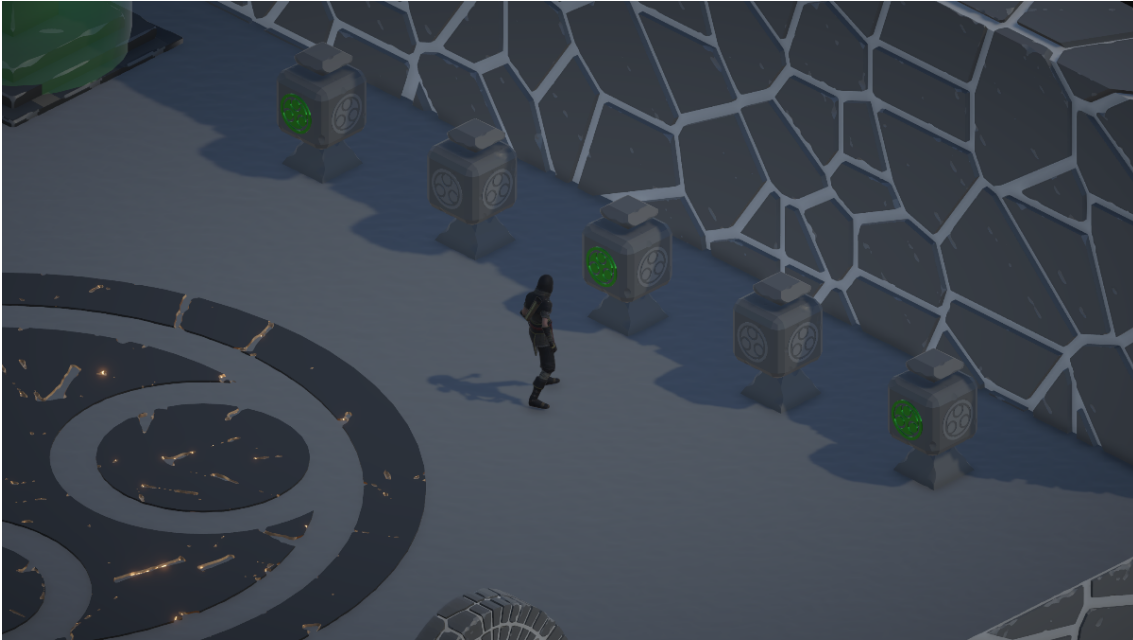


Figura 16. Blend tree

El tercer puzzle tracta d'unes plaques a pressió que, quan detecten al jugador, aquestes es tornen en un estat actiu i el jugador ha de passar per sobre de totes elles i deixar-les totes en l'estat actiu per solucionar-lo. Aquest puzzle també està inspirat per un puzzle del videojoc Genshin Impact, enllaço el vídeo on es pot veure el puzzle [aquí](#).

Per crear aquest puzzle també es va utilitzar una matriu, que guarda cada una de les plaques a un índex. Aquestes si detecten al jugador i estaven desactivades, s'activen, en canvi si ja estaven activades i el jugador torna a passar per sobre, es desactiven. El puzzle comprova per cada canvi si totes les plaques tenen l'estat actiu i si és així el puzzle es dona per solucionat i la cúpula de la gemma desapareix.

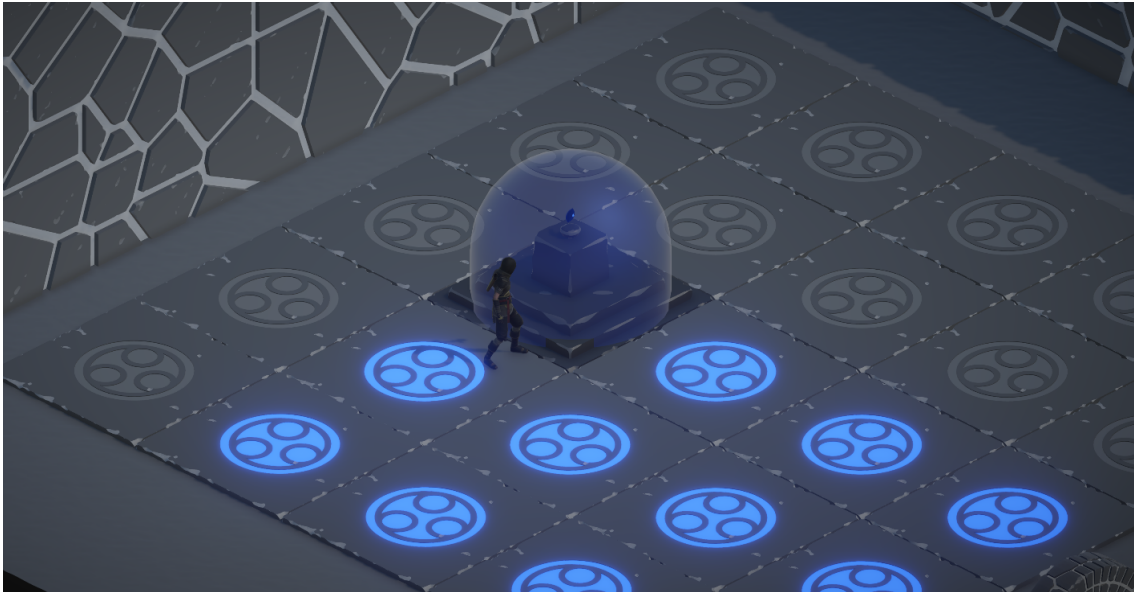


Figura 16. Blend tree

La porta té la funció de comprovar si totes les gemmes estan renderitzades, ja que cada vegada que el jugador ha anat agafant cada gema, s'ha renderitzat una copia seva a la porta, per tant si totes estan renderitzades i el jugador les activa, la porta s'obre deixant passar al jugador cap a l'escena final.



Figura 16. Blend tree

## Creació de la UI

La creació de la interfície d'usuari va ser de les últimes coses a implementar. Es van crear els menús d'inici, una breu introducció perquè el jugador entengui la història, el menú de pausa i la pantalla final que es mostra quan has completat el joc. La navegació d'aquests menús és molt bàsica, ja que utilitzen botons i *scripts* molt senzills per carregar escenes i mostrar/ocultar el menú.

## Texturització i il·luminació

La texturització i la il·luminació a Unity va ser una mica limitada, ja que moltes de les textures que a Blender es veien molt bé quan es van passar a Unity. Potser és un dels apartats en els que es voldria haver treballat una mica més però el desenvolupament dels punts anteriors va ser una mica més demandant del que s'havia previst.

## 5.3 Postproducció

En aquest apartat es descriuran els processos duts a terme un cop acabat el procés de producció.

En aquest cas els que es va realitzar un cop acabat el procés de producció és la implementació de l'àudio i un petit retoc en la il·luminació global de l'escena.

L'àudio es va treure de fons que permeten el seu ús sempre i que no sigui comercial.

També es va posar un fons en negre ja que Unity per defecte posa un *Skybox* com a fons.

## 6. Validació del projecte

De cara a realitzar la validació del projecte, per anar controlant que el projecte avançava favorablement i per tal de rebre *feedback*, a la gestió del projecte es van programar una sèrie de tests interns i externs.

Els tests interns es van anar fent cada cop que hi havia un avenç substantiu en la programació del joc. Això vol dir que cada cop que es va programar una funcionalitat nova, ja sigués un puzzle, un canvi d'escena o l'obertura de la porta final, es va testejar a fons per comprovar la seva correcta implementació dins del joc i revisar que no havia afectat al funcionament de cap altre funcionalitat.

Un cop el joc havia quedat completat i era plenament jugable (encara sense haver fet els menús ni haver posat audio) es va procedir a realitzar els primers tests externs. Aquests tests es van fer amb jugadors voluntaris a través de la plataforma de Discord. El procediment que es va seguir era el següent: se'ls enviava una còpia de la *build* del videojoc i es realitzava una trucada mentre compartien pantalla, per tal de poder fer un seguiment de com van jugant, i anaven comentant els aspectes que veien que els agradaven o que canviarien.

Aquests primers tests externs van anar molt bé ja que, gràcies als jugadors, es van poder identificar dos errors que arruïnaven l'experiència del jugador i, en un dels casos, feia que el joc no es pogués superar ja que un dels puzzles es tornava impossible de resoldre.

El primer error era que l'altar de la gemma d'ametista estava massa a prop de les parets de la sala, per tant quan el jugador, que encara no havia solucionat el puzzle per desbloquejar aquella gemma, volia passar entre la cúpula que impedia que pogués agafar la gemma i les parets de la sala, el personatge travessava per uns pocs frames la cúpula i, si era molt precís i aconseguia apretar la tecla E en aquell precís instant, el jugador podia agafar la gemma sense haver necessitat solucionar el puzzle.

Això es va solucionar separant l'altar de les parets el suficient com perquè el jugador pogués passar sense problemes pel darrera.

El segon error era el més crític, i és que en el puzzle dels cubs que giren, si donaves l'ordre a un cub que girés i anaves ràpid al del costat i li donaves l'ordre també, sense haver esperat a que la primera acció de rotació s'hagués acabat, la suma de les ordres

feia entrar en conflicte alguns cubs i quedaven mal orientats, a vegades en diagonal. Cosa que feia que el puzzle fós impossible de solucionar.

Aquest error es va solucionar fent un repàs a l'script i canviant el sistema com funcionava. Es va fer de tal manera que mentre un cub estigués rotant, es desactivaven els *box collider* que detectaven i activaven la rotació dels cubs quan el jugador estava a dins quan els cubs estaven girant. D'aquesta manera el jugador no podia donar una segona ordre de rotació a un altre cub mentre encara s'estigués realitzant l'acció de rotació del cub anterior.

A part d'aquests dos errors el feedback va ser bo, van comentar que el joc es veia bé i fluït, els puzzles eren lògics i els controls eren els adients.

Pels segons tests externs es va utilitzar el mateix mètode, una trucada a través de Discord i amb la pantalla compartida. Aquest cop la build era una més completa, ja que incorporava els canvis i millores fets i també noves funcionalitats com la pantalla inicial, el menú i els sons d'ambientació, apart d'una millora en el texturitzat d'algunes escenes.

Aquests segons tests van anar genial, el joc funcionava a la perfecció, sense cap error ni bug inesperat, els jugadors van saber solucionar els puzzles intuitivament però no van trobar la solució de manera fàcil.

Com a aspectes que van comentar, la possible ampliació del joc amb més puzzles i la possibilitat de que el personatge es mogués més ràpid, a part d'això van trobar el joc molt ben fet i atractiu.

## 7. Conclusions

El resultat obtingut ha estat molt satisfactori al final de tot, ja que crear un videojoc una persona sola és una tasca molt complicada i requereix de moltes hores i esforç.

D'aquest treball he après que gestionar bé des d'un inici un projecte tan gran com la creació d'un videojoc pot salvar moltes hores a la producció d'aquest.

Unity és un programa molt potent, ja ho ha estat sempre però visualment sempre ha estat darrera d'altres motors com Unreal Engine. Aquest treball m'ha ensenyat que amb una bona programació de shaders es poden aconseguir uns resultats molt satisfactoris visualment. Hi ha una altra versió del motor de render que és l'HDRP que és molt capaç de competir amb Unreal Engine en l'aspecte visual.

M'ha ensenyat molt a millorar com a programador de videojocs, ja que m'he topat amb mil i un problemes i m'ha fet pensar varies solucions a un sol problema i mirar d'implementar-les, m'ha ensenyat a pensar més lògicament de manera senzilla i simple.

### 7.1 Línies de futur

El futur d'aquest projecte podria ser l'ampliació del nombre de puzles i la creació de nous camins i sales que convidin al jugador a explorar més. L'aspecte visual també és una cosa a tenir en compte ja que és un aspecte que no s'ha pogut tocar com es volia.

## 8. Bibliografia

- a good boy games. (2021, October 25). *Isometric Camera in Unity | Bite-Sized Tutorials* [Video]. YouTube. <https://www.youtube.com/watch?v=c8-l-1aMVjk>
- Alexdrummond. (n.d.-a). *Blood Crystal Prison - [isometric] [Battlemap] [68x53]* : r/FantasyMaps.  
[https://www.reddit.com/r/FantasyMaps/comments/vbucxz/blood\\_crystal\\_prison\\_isometric\\_battlemap\\_68x53/](https://www.reddit.com/r/FantasyMaps/comments/vbucxz/blood_crystal_prison_isometric_battlemap_68x53/)
- Alexdrummond. (n.d.-b). *Dwarven Mines "The Forge" - [isometric] [Battlemap] [68x53]* : r/DungeonMasters.  
[https://www.reddit.com/r/DungeonMasters/comments/vacyns/dwarven\\_mines\\_the\\_forge\\_isometric\\_battlemap\\_68x53/](https://www.reddit.com/r/DungeonMasters/comments/vacyns/dwarven_mines_the_forge_isometric_battlemap_68x53/)
- Alley Viper. (2012, February 20). *Uncharted 3 La Traición de Drake - Puzle Château n° 1* [Video]. YouTube.  
<https://www.youtube.com/watch?v=rF4VvZNG5og>
- Amagammon. (n.d.). *Blender 2.5 Realistic gemstones (part 1) Emerald Cut Gemstone* [Video]. YouTube.  
[https://www.youtube.com/watch?v=V2mE3Z8CeVc&list=PL\\_D\\_fk\\_I0tixzRPJS17pIvJNQNusu0cAY&index=27&t=310s](https://www.youtube.com/watch?v=V2mE3Z8CeVc&list=PL_D_fk_I0tixzRPJS17pIvJNQNusu0cAY&index=27&t=310s)
- *Ancient Sky* by VictorWayne — *Royalty Free Music Download*. (n.d.).  
<https://tunetank.com/track/4211-ancient-sky/>
- Brackeys. (2017, November 29). *START MENU in Unity* [Video]. YouTube.  
[https://www.youtube.com/watch?v=zc8ac\\_qUXQY](https://www.youtube.com/watch?v=zc8ac_qUXQY)

- Brazie, A. (2023, June 9). How to Make a Functional Game Design Document (Examples and Templates). *Game Design Skills*.  
<https://gamedesignskills.com/game-design/document/>
- Don Pachi. (n.d.). YouTube. <https://www.youtube.com/@DonPachi/videos>
- Don Pachi. (2019, June 30). *Unity 2019: Importar y animar un personaje 3D* [Video]. YouTube. [https://www.youtube.com/watch?v=Ay\\_oy6GXC-s](https://www.youtube.com/watch?v=Ay_oy6GXC-s)
- *ed42f901ffa5fa27f1c0433779f0054e.png (1920×1080)*. (n.d.).  
<https://i.pinimg.com/originals/ed/42/f9/ed42f901ffa5fa27f1c0433779f0054e.png>
- Enma. (2022, March 30). *[The Chasm] Light-Up Tile Puzzle Solution | Genshin Impact* [Video]. YouTube. <https://www.youtube.com/watch?v=IcGtkF3SdYY>
- FlipCard. (2021, July 23). *5 Cubes Puzzle Water Underground Araumi Inazuma Part 1 & Part 2 SOLVED! | Genshin Impact* [Video]. YouTube.  
<https://www.youtube.com/watch?v=NFIjpadrJp0>
- Games, A. G. B. (n.d.). *Smooth Camera Follow in Unity | Bite-Sized Tutorials* [Video]. YouTube.  
[https://www.youtube.com/watch?v=ZPUtQ4pGGWs&list=PL\\_D\\_fk\\_IOTixzRPJS17pIvJNQNusu0cAY&index=17](https://www.youtube.com/watch?v=ZPUtQ4pGGWs&list=PL_D_fk_IOTixzRPJS17pIvJNQNusu0cAY&index=17)
- iHeartGameDev. (2021, January 17). *How to Move Characters In Unity 3D | Character Controllers Explained* [Video]. YouTube.  
<https://www.youtube.com/watch?v=e94KggaEAr4>
- Mina Pêcheux. (2022, March 9). *Configure & animate Mixamo characters [Unity/C# tutorial]* [Video]. YouTube.  
<https://www.youtube.com/watch?v=8Pk7FI629O8>
- Mixamo. (n.d.). <https://www.mixamo.com/#/>



- *Premium AI Image | Isometric game battlemap for videogames and table top RPGs.* (n.d.-a). Freepik.  
[https://www.freepik.com/premium-ai-image/isometric-game-battlemap-videogames-table-top-rpgs\\_32638007.htm](https://www.freepik.com/premium-ai-image/isometric-game-battlemap-videogames-table-top-rpgs_32638007.htm)
- *Premium AI Image | Isometric game battlemap for videogames and table top RPGs.* (n.d.-b). Freepik.  
[https://www.freepik.com/premium-ai-image/isometric-game-battlemap-videogames-table-top-rpgs\\_32638006.htm](https://www.freepik.com/premium-ai-image/isometric-game-battlemap-videogames-table-top-rpgs_32638006.htm)
- Rohrbach ® the Jewelry Jedi. (2018, May 25). *VIDEO 21 Diseño de joyería 3D BLENDER Escuela de joyería CADJ* [Video]. YouTube.  
<https://www.youtube.com/watch?v=TyCWGYMimNw>
- Rohrbach ® the Jewelry Jedi. (2020, May 18). *Gemstones TUTORIAL 3 Jewelry 3D design with Damien Rohrbach and Blender* [Video]. YouTube.  
<https://www.youtube.com/watch?v=HP6ob6uLpBY>
- Runway, P. (n.d.). *Crystal Desert Blender Tutorial with Commentary* [Video]. YouTube.  
[https://www.youtube.com/watch?v=1TN\\_d0-azLU&list=PL\\_D\\_fk\\_IOTixzRPJS17pIvJNQNusu0cAY&index=32&t=1612s](https://www.youtube.com/watch?v=1TN_d0-azLU&list=PL_D_fk_IOTixzRPJS17pIvJNQNusu0cAY&index=32&t=1612s)
- Solarmusic. (2022, July 11). *Dripping Water in Cave | Royalty-free Music.* Pixabay. <https://pixabay.com/sound-effects/dripping-water-in-cave-114694/>
- Tarodev. (2021, October 26). *Isometric Character Controller in Unity* [Video]. YouTube. <https://www.youtube.com/watch?v=8ZxVBCvJDWk>
- Technologies, U. (n.d.). *Unity - Manual: Unity User Manual 2022.3 (LTS).* <https://docs.unity3d.com/Manual/index.html>

- *UGuruz*. (n.d.). YouTube. <https://www.youtube.com/@UGuruz/videos>
- *Unity y Mixamo*. (n.d.). YouTube.

<https://www.youtube.com/playlist?list=PLTUZMVrECom01n8VqvqiPYQlvYja>

YoZWb

## 9. Annexos

### Exemples

Guió Literari

Guió Tècnic

GDD