



This is the **published version** of the bachelor thesis:

Blanco Prous, Abel; Bernal del Nozal, Jorge, tut. Desenvolupament d'un videojoc estil roguelike 2D amb generació procedural de sales. 2025. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/308764>

under the terms of the  license

Desenvolupament d'un videojoc estil roguelike 2D amb generació procedural de sales

Abel Blanco Prous

6 de febrer de 2025

Resum— En aquest projecte s'ha desenvolupat un videojoc 2D de gènere roguelike amb generació procedural de sales utilitzant el motor de desenvolupament de jocs Godot. La generació de sales es fa a partir d'una sèrie de sales predissenyades; en generar una nova sala es tria aleatòriament una de les sales del conjunt predissenyat. El jugador decideix, a cada sala, quina direcció vol prendre i, així, decideix de manera implícita la configuració final de sales del joc. El jugador també pot decidir quines millores vol agafar, canviant el desenvolupament de la partida i la estratègia de joc. Com a resultats s'ha aconseguit un joc completament funcional i configurable on el jugador ha de travessar un conjunt de sales amb enemics fins a arribar a una sala final on ha de derrotar un enemic final mantenint-se en vida tota la partida.

Paraules clau— Videojoc Roguelike, Generació Procedural, Godot, Sales Predissenyades, Màquina d'estats, Diseny de Nivells, Selector de Dificultat.

Abstract— In this project, a 2D roguelike video game with procedural room generation has been developed using the Godot game development engine. Room generation is done from a series of pre-designed rooms; when generating a new room, one of the rooms from the pre-designed set is randomly chosen. The player decides, in each room, which direction he wants to take and, thus, implicitly decides the final room configuration of the game. The player can also decide which improvements to take, changing the development of the game and the game strategy. As a result, a fully functional and configurable game has been achieved where the player must cross a set of rooms with enemies until reaching a final room where he must defeat a final enemy while staying alive throughout the game.

Keywords— Roguelike Video Game, Procedural Generation, Godot, Pre-designed Rooms, State Machine, Level Design, Difficulty Selector.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

ELS roguelike són un subgènere dels videojocs de rol caracteritzats per aventures procedurals que transcorren en laberints de cambres generades aleatòriament i que es reconfiguren en cada partida [1]. El nom d'aquest gènere prové del joc *Rogue* de l'any 1980 (Roguelike ve de "jocs similars a Rogue"), desenvolupat per Michael Toy i Glenn Wichman. A la figura 1 es mostra un exemple de partida d'aquest joc. Tot i que *Rogue* dona nom al gènere, els videojocs *Star Trek, a simple explora-*

tion/strategy game (desenvolupat per Mike Mayfield i Bob Leedom, 1971) i *Beneath Apple Manor* (desenvolupat per Don Worth, 1978) ja van introduir mecàniques pioneres i temàtiques que més tard es consolidarien en els roguelike moderns.

Rogue utilitzava caràcters ASCII com a gràfics i la seva trama era simple: un aventurer anònim, simbolitzat per "@", explora un laberint a la recerca de l'Amulet de Yendor, trobant-se amb monstres representats per lletres de l'alfabet (per exemple, "E" per a Emu i "S" per a Serp). A cada mort del jugador, tant els nivells del laberint com les propietats dels objectes es regeneraven de manera totalment aleatòria, obligant als jugadors a adaptar-se a noves condicions en cada partida. [2].

L'any 2008, durant la *International Roguelike Development Conference*, es va establir el nom del gènere roguelike provenint del joc *Rogue*, definint en la "Berlin Interpretation" (La interpretació de Berlín) el que era un joc roguelike,

• E-mail de contacte: abel.blanco@autonoma.cat
• Menció realitzada: Computació
• Treball tutoritzat per: Jorge Bernal del Nozal (Ciències de la Computació)
• Curs 2024/25

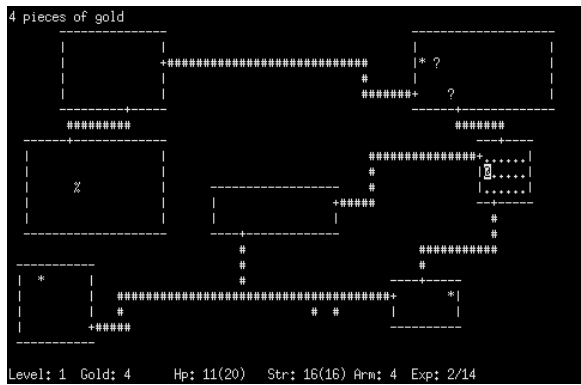


Fig. 1: Captura d'una partida de Rogue

definint com a criteris, entre altres, que els roguelike eren jocs dividits per cambres que utilitzin un sistema de cases, que es desenvolupin per torns i amb mort permanent. Cal destacar que, les definicions oficials de gèneres no existeixen en la indústria dels videojocs sinó que surten de l'ús comú [2-3].

Així doncs, aquest projecte consistirà en el desenvolupament d'un joc 2D del gènere Roguelike, utilitzant el motor Godot [4]. El joc utilitzarà tècniques de generació procedural per generar les diferents cambres de forma dinàmica per les que avançarà el protagonista. Això augmentarà la rejugabilitat del joc i oferirà una experiència diferent en cada partida. Les sales inclouran diferents enemics controlats per una màquina d'estats finits. En ser derrotats, s'obriran d'1 a 3 portes en direccions diferents per la que s'hagi accedit a la sala prèviament, per les que el jugador continuarà la partida. Aquest mecanisme introdueix aleatorietat en la distribució de sales a cada partida, ja que la combinació de portes obertes alterarà el camí disponible

A part de les sales generades proceduralment hi hauran sales especials no generades d'aquesta manera. Una d'aquestes serà la sala inicial on simplement es generaran portes perquè el jugador comenci l'aventura, una sala que contindrà una font per restaurar vida o millorar armament i la sala final on el protagonista s'enfrontarà a un enemic final. Per acabar, també es podrà ajustar el nivell del joc en tres nivells: fàcil, normal i difícil, incrementant el nombre de sales generades proceduralment abans de l'enfrontament contra l'enemic final.

2 OBJECTIUS DEL PROJECTE

L'objectiu final del projecte és crear el joc de gènere roguelike especificat anteriorment, per aconseguir això, desglossem el projecte en una sèrie d'objectius a complir:

- Aprendre a utilitzar la game engine Godot
- Recercar algoritmes de Generació Procedural per nivells de videojocs
- Establir el mètode de generació de les cambres
- Realitzar el disseny gràfic del joc
- Crear la 'IA' dels enemics (màquina d'estats)
- Crear un menú per escollir la dificultat
- Provar i polir el joc per posar-ho a punt

3 ESTAT DEL ART

En l'indústria dels videojocs podem trobar diversos exemples de videojocs similars al projecte que es vol realitzar. Un dels més populars és *The Binding of Isaac*, un roguelike 2d de vista zenital on les sales per les quals transita el jugador són generades proceduralment i es permetent retornar a zones prèvies. Durant el recorregut, s'enfronta amb diversos enemics, pot trobar claus per obrir cofres, armes i millores que li són d'utilitat i s'acaba enfrontant amb enemics finals [5].

Un altre exemple és el joc *Hades*, un roguelike 2D amb vista isomètrica (similar a la vista zenital) on les càmeres es generen de forma procedural: es seleccionen d'un conjunt predissenyat, però es combinen de manera aleatòria per formar els nivells mentre el protagonista intenta escapar de l'inframón. Cada sala està connectada de manera lineal i no s'hi pot tornar enrere. Durant el recorregut, el jugador s'enfronta a enemics, obté recompenses com millores d'habilitats o benediccions dels déus olímpics, i, al final de cada àrea, s'enfronta a enemics finals abans de poder avançar [6]. A la figura 2 es mostra una partida d'*Hades*.

Per afegir un últim exemple, també trobem el joc *Monolighter*, un altre joc 2d amb vista zenital. Tot i que no és completament roguelike té dues parts: durant el matí del joc el protagonista s'ocupa d'una tenda mentre que a la nit explora masmorres generades de forma procedural. Les habitacions de les masmorres es connecten aleatòriament i estan plenes d'enemics que protegiran cofres amb objectes valuosos. Aquests objectes es fan servir per gestionar la botiga. A mesura que s'avança per les masmorres, es poden trobar millores, trampes i enfrontar-se a enemics al final de cada masmorra. Durant la fase de masmorres, no es pot tornar a les habitacions prèvies i els objectes es perden si el jugador mor [7].

Hi ha diverses solucions en utilitzar la generació procedural per crear masmorres en jocs [8]. Una solució són les *gramàtiques generatives*, que són regles formals (similars a les dels llenguatges de programació) utilitzades per generar estructures de masmorres seguint patrons predefinits. D'aquí en surten variacions com la gramàtica de grafs, que en lloc de construir frases lingüístiques, construeix grafs on els símbols no són paraules sinó nodes i arestes per representar la connexió entre habitacions [9].

Una altra variació és la gramàtica de formes, on les regles s'utilitzen per generar estructures dins les habitacions i els símbols que s'utilitzen són formes geomètriques [10] [11].

Una altra opció és el mètode d'*autòmats cel·lulars*, que

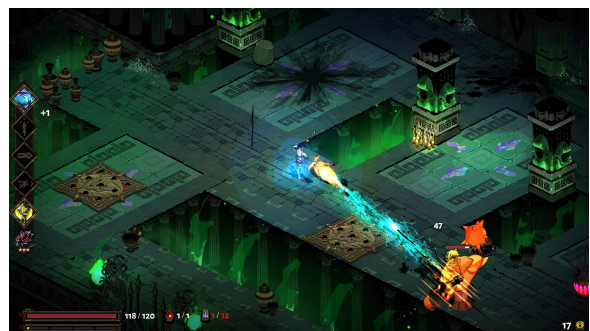


Fig. 2: Captura d'una partida d'Hades

emula un sistema de cel·les que evolucionen segons regles locals per crear patrons espacials complexos que poden representar les caveres o masmorres irregulars amb una sensació caòtica i natural. Aquest mètode permet generar mapes en temps real, però no assegura la connexió entre totes les àrees accessibles. Un bon exemple ho trobem al treball de Johnson [12], que utilitza les capacitats d'autoorganització dels automats cel·lulars per generar nivells de coves. Aquest treball defineix el veïnatge d'una cel·la com les vuit cel·les que l'envolten (veïnatge Moore), on els possibles estats són terra, roca, mur i defineix unes regles per decidir cada cel·la quin estat serà. El conjunt de regles s'aplica de manera iterativa en diverses generacions.

Els *algorismes genètics* utilitzen l'evolució i la selecció natural per generar masmorres optimitzades després de diverses generacions a través de la mutació (petits canvis) i l'encreuament de "gens" (característiques) de dissenys de nivells previs. Aquesta tècnica requereix una funció per avaluar la qualitat de cada disseny generat, seleccionant i combinant els millors per crear-ne de nous. Amb un procés iteratiu, les masmorres van evolucionant cap a formacions més adequades, assegurant que es compleixin certs criteris que s'indiquin, com la distribució d'habitacions i passadissos o punts de control per a una millor experiència de joc. Diversos autors com Hartsook et al. [13], Valtchanov and Brown [14] i Ashlock et al. [15] han aplicat algorismes genètics en la generació procedural de nivells.

Per últim, comentar *els mètodes basats en restriccions*, que generen masmorres respectant restriccions específiques, assegurant que les solucions compleixin amb criteris predefinitos. Per exemple, Roden i Parberry [16] proposen un enfocament on es genera un graf 3D no dirigit que representa l'estructura del nivell. Restringint tant la topologia del graf com les propietats dels nodes, es controla la distribució espacial de les zones del nivell, la connexió entre elles i l'ordre en què el jugador pot accedir-hi.

4 METODOLOGIA I PLANIFICACIÓ

El desenvolupament d'aquest projecte seguirà una metodologia *Kanban* [17]. Aquest és un mètode visual de gestió de projectes que es basa en un tauler organitzat per columnes que representen l'etapa de treball de la tasca, que poden ser: To do, On going, Done, i unes targetes que són les tasques a realitzar, que avancen per les diferents columnes fins que es finalitza la tasca.

Per gestionar el seguiment del projecte s'utilitzarà la plataforma en línia *Trello* [18], la qual implementa el mètode Kanban, per facilitar l'organització del treball.

La planificació temporal de les tasques a realitzar es pot dividir en 4 fases (les tasques i la seva durada estan representades al Diagrama de Gantt del projecte a l'apèndix). Aquestes fases són:

1. Fase inicial (del 10/09/2024 fins al 06/10/2024): Aquesta fase es centra en la recollida d'informació, definir els objectius del projecte i completar l'informe inicial.
2. Fase d'implementació (del 07/10/2024 fins al 15/12/2024): Aquesta és la fase més duradora del projecte on es desenvoluparà el videojoc i consta en

decidir l'algoritme de generació de les diferents sales, programar el comportament i generació dels enemics i polir el joc per complir amb els objectius establerts. Es faràn les entregues dels documents de progrés que facin falta.

3. Fase de documentació (del 16/12/2024 fins al 04/02/2025): S'analitzaran els resultats obtinguts i s'acabarà de documentar el treball fet.
4. Fase final (del 05/02/2025 fins al 17/02/2025): S'acabarà de fer les entregues del dossier, es farà la proposta de la presentació i es presentarà el treball.

5 REQUISITS HARDWARE/SOFTWARE

Aquest projecte es desenvoluparà en un ordinador de sobretaula utilitzant Godot. S'ha escollit Godot en lloc d'altres motors pel que el seu aprenentatge és més accessible i per la seva comunitat activa, comparable a la de Unity, l'altre motor considerat. Godot és més lleuger, consumeix menys recursos i està més especialitzat en 2D, oferint eines específiques que faciliten el desenvolupament. A més, utilitza *GScript*, un llenguatge inspirat en *Python*, amb el que s'està més familiaritzat que *C#*, el llenguatge usat en Unity. Els requisits mínims de l'editor per PC són [19]:

CPU	Windows: CPU x86_32 amb SSE2, qualsevol CPU x86_64, CPU ARMv8 macOS: CPU x86_64 o ARM (Apple Silicon) Linux: CPU x86_32 amb SSE2, CPU x86_64, ARMv7 o ARMv8
GPU	Mode Forward+: Gràfics integrats amb Vulkan 1.0 Mode Compatibility: Gràfics integrats amb OpenGL 3.3
RAM	4 GB
Espai	200 MB per l'executable i 1.3 GB més per exportar projectes
OS	Windows 7, macOS 10.13 o més nou, Linux post-2016

6 RISCOS DEL PROJECTE

Cas	Risc	Contingència	Probabilitat
L'algoritme de generació no funciona	Alt	Recercar i escollir altres algorismes	Mitjana
Dificultat al crear els enemics	Mitjà	Canviar la base de l'implementació dels enemics	Baixa
Planificació temporal no seguida	Mitjà	Replanificar les dades i retocar els objectius si cal	Mitjana

7 DESENVOLUPAMENT

7.1 Escenes

El joc està format per diferents escenes de Godot, ajuntades en una escena principal. Per parlar de les escenes dissenyades en aquest projecte, hem de parlar primer sobre que són les escenes en Godot i quina és la seva filosofia a l'hora de desenvolupar un joc per entendre com s'ha desenvolupat aquest projecte.

En Godot, la idea és dividir el joc a fer en diferents escenes que pots reutilitzar [20]. Les escenes estan formades per nodes, que són els blocs de construcció més petits del joc. Aquests nodes els proveeix Godot i es combinen per formar les escenes. Per exemple, el node KinematicBody2D s'utilitza per representar un cos 2D i gestionar la seva física de col·lisions i els seus moviments. Ajuntant aquest node amb els de Sprite, Camera2D i CollisionShape2D podríem tenir una escena senzilla d'un jugador.

Els nodes s'ordenen en forma d'arbre (com es veu en la figura 3), formant així una escena.

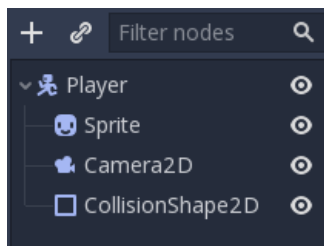


Fig. 3: Captura de mostra d'una escena jugador simple

Una escena pot ser un personatge, una arma, un menú o un nivell complet. Qualsevol cosa pot ser definida com a escena, ja que la seva naturalesa és molt flexible. Aquestes escenes es poden introduir dintre d'altres escenes per aconseguir l'efecte desitjat, com per exemple, dissenyar l'escena de jugador i afegir-la a l'escena d'un nivell complet.

Totes les escenes s'acaben ajuntant en una principal, que és l'arbre d'escena. Ho podem veure en la figura 4.

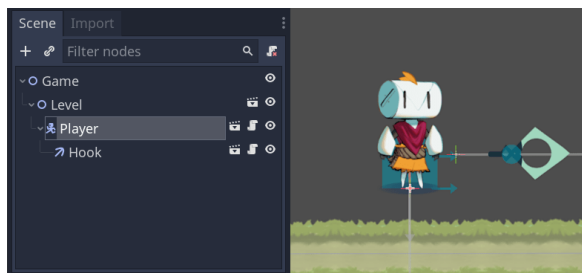


Fig. 4: Captura de mostra al juntar escenes, de la documentació de Godot

Així doncs en el projecte hi ha les escenes del jugador, la porta, les escenes d'enemics i les escenes de les sales, com es pot veure a la figura 5 [21-39].

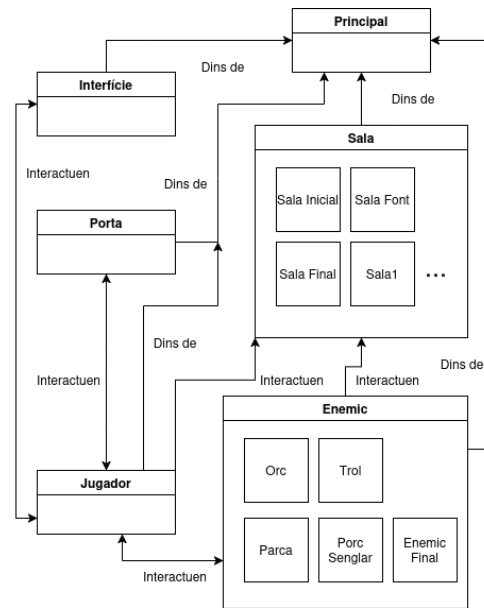


Fig. 5: Relació de les escenes

Aquestes són les diferents escenes plantejades al joc:

- L'escena del jugador s'encarrega de contenir el seu sprite, gestionar el seu desplaçament (el moviment es realitza en 4 direccions) i les accions d'atac.
 - L'escena de la porta conté el seu sprite i s'encarrega d'informar que el jugador passa per la porta.
 - Les escenes dels enemics contenen els seus sprites i gestionen el seu comportament mitjançant una màquina d'estats.
 - Respecte a les escenes de les sales, en tenim quatre tipus: Sala inicial, sala predissenyada (les sales intermèdies), sala font i sala final. Cada una compta amb els seus sprites.
- La sala inicial és una sala sense obstacles. Les diferents sales predissenyades contenen obstacles, amb diverses roques col·locades en diferents posicions. La sala font té una font al mig, que servirà per millorar el personatge, i la sala final és on transcorre la batalla definitiva.
- Per últim, l'escena de l'interfície és la que s'encarrega de mostrar la barra de vida del jugador, del enemic final (quan toca), el tipus d'enemic final i els atributs del jugador: Nombre de millores d'atac bàsic, nombre de punts d'atac a distància i indicador d'atac a distància disponible.

7.2 Partida

Com s'ha explicat abans, a l'escena principal es criden les altres escenes. El personatge comença en la sala inicial on es generen d'1 a 4 portes en les parets superior, inferior i laterals. En entrar per una de les portes, s'elimina la sala actual amb les portes generades i s'actualitza el comptador que controla quantes sales intermèdies hi ha abans d'una sala font (el nombre canviarà segons la dificultat). Seguidament, es genera una sala intermèdia predissenyada aleatòriament (com es veu d'exemple en la figura 6) o, en cas



Fig. 6: Captura d'una sala intermèdia d'exemple

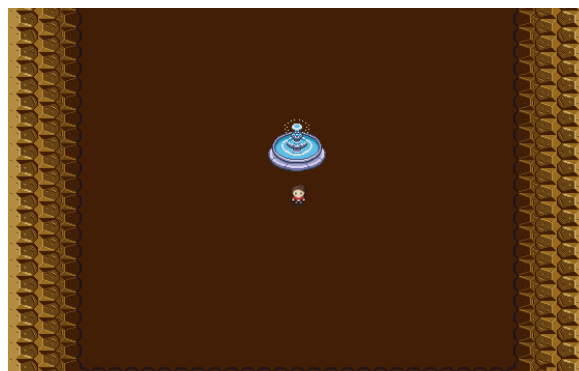


Fig. 8: Captura de la sala font

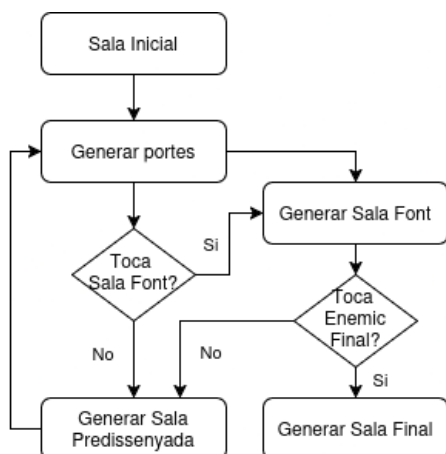


Fig. 7: Diagrama generació sales

que toqui, la sala font. Aquest procés, il·lustrat en la figura 7, es va repetint fins que s'arriba a un nombre de sales fonts generades (canviarà segons la dificultat), on després de la sala font es generarà la sala final.

A partir de la sala inicial, el nombre de portes que es generaran serà d'1 a 3 i el jugador es posicionarà a la zona oposada de la nova sala respecte a la porta per la qual ha entrat. Així, si el jugador entra per la porta inferior, en la següent sala apareixerà a la zona superior. El disseny de les sales predissenyades no es repetirà en canviar de sala.

El joc consistirà en avançar per les diferents sales i en travessar un cert nombre de sales (que canviarà segons una selecció de dificultat a l'inici de cada partida) fins arribar a una sala final on el jugador s'enfrontarà a un enemic final. Cada nova partida que jugui el jugador serà diferent segons les millores que aquest esculli, les sales a les quals s'enfronti i el tipus d'enemic final d'aquella partida en concret.

A l'inici de cada partida, s'informarà el jugador del tipus d'enemic final al qual s'enfrontarà, que pot ser de 3 tipus: Foc, Aigua o Planta. En la partida el jugador podrà optar per obtenir un atac a distància poderós (un làser màgic) que pot ajudar-lo a derrotar l'enemic final amb més facilitat. Cada tipus d'enemic final té debilitat a un altre tipus: Foc és dèbil a Aigua, Aigua és dèbil a Planta i Planta és dèbil a Foc. Cada sala a la que s'enfronti el jugador també serà d'un d'aquests tipus, el que serveix per determinar si es pot obtenir l'atac a distància en les sales font.

En arribar a una Sala Font, que es veu a la figura 8, les opcions de millora que tindrà el jugador seran restaurar vida perduda, millorar el mal dels atacs bàsics o obtenir un punt d'acumulació per l'atac a distància, la disponibilitat

d'aquesta millora depèn de les sales travessades.

Cada vegada que s'arriba a una sala font es revisa si la majoria de les sales per les quals ha passat el jugador des de l'última sala font (o sala inicial la primera vegada) són del tipus contrari a l'enemic. Per exemple, si la majoria de les sales travessades han sigut de tipus aigua i l'enemic final és de tipus foc, hi haurà l'opció d'aconseguir un punt per desbloquejar l'atac a distància. Respecte a aquest, seran necessaris 3 punts per desbloquejar-ho.

Un cop desbloquejat l'atac a distància, cada 10 atacs bàsics encertats es podrà llançar l'atac a distància. Dependent de la selecció de dificultat, es pot obtenir el làser abans de la sala final. El làser farà menys mal als enemics de les sales (10 de dany), mentre que li farà més mal a l'enemic final (30 de dany).

Per tant, si el jugador vol optar per obtenir l'atac a distància, haurà de planificar les sales per les quals passarà. Haurà de tindre en compte que els enemics de les sales del tipus contrari a l'enemic final, tenen una mica més de vida i velocitat.

7.3 Jugador

Per controlar el moviment del jugador, com es veu en la figura 9, es fan servir les fletxes del teclat, per atacar cos a cos la tecla Z i per llançar el làser quan està disponible la tecla X. Els atributs del jugador són:

Vida	Atac bàsic	Atac làser	Dany extra
60	3	10	20



Fig. 9: El jugador. A l'esquerra caminant/parat i a la dreta atacant

Els números d'atac i dany extra són els valors que es resten als enemics, recordant que el dany extra només s'aplica per l'enemic final. El dany de l'atac làser a l'enemic final és 30 en total.

7.4 Enemics

En l'interior de les sales intermèdies el jugador es trobarà amb enemics que l'intentaran derrotar. El comportament de cada enemic està representat per una màquina d'estats, que és un model que s'utilitza per representar sistemes mitjançant estats (una situació en específic i la que el sistema, en aquest cas l'enemic, pot estar) i transicions (el canvi d'un estat a un altre quan es compleix una condició).

Cada enemic també compta amb un con de visió (varia segons l'enemic) que s'utilitza per la detecció del jugador. Cadascun està creat amb una escena de Godot i les escenes dels enemics es troben dins les escenes de les sales.

La màquina d'estats, il·lustrada en la figura 10, de cada enemic té 5 estats: *Idle*, *Chase*, *Attack*, *Scape* i *Death*.

- **Estat Idle:** En aquest estat l'enemic tria una direcció aleatòria i camina cap a ella fins a trobar un obstacle o fins que passa un interval de temps aleatori (entre 1.5 i 2 segons). Seguidament, para de moure's un interval de temps aleatori (entre 0.5 i 0.8 segons) i torna a repetir el procés. Si en algun moment el jugador entra en el con de visió de l'enemic o està molt a prop d'aquest, l'enemic passa a estat *Chase*. Si la vida de l'enemic és baixa, es passarà a estat *Scape*.
- **Estat Chase:** En aquest cas, l'enemic es mourà cap a la direcció del jugador evitant els obstacles que es trobi pel camí. Quan estigui prou a prop del jugador perquè el seu atac encerti, es passarà a l'estat *Attack*. Si la vida de l'enemic és baixa, es passarà a estat *Scape*.
- **Estat Attack:** En aquest estat, l'enemic realitza l'atac al jugador. Si encerta, li disminueix vida al jugador, espera 0.2 segons i, en cas que el jugador continuï estant en rang d'atac, torna a atacar-ho. Es repeteix el procés fins que el jugador ja no estigui en rang d'atac, en el qual cas es retornarà a l'estat *Chase* o a l'estat *Scape* si té poca vida.
- **Estat Scape:** En aquest estat l'enemic s'allunyarà del jugador mentre tingui poca vida. Quan estigui a certa distància del jugador, parerà de moure's i recuperarà la meitat de la vida. Seguidament, passarà a estat *Chase*.
- **Estat Death:** Aquest estat s'encarrega de la mort de l'enemic, és a dir, de l'animació final de mort i d'alliberar l'espai que ocupava. **Quan en qualsevol estat la vida descendeixi per sota el 0, s'anirà a aquest estat.**

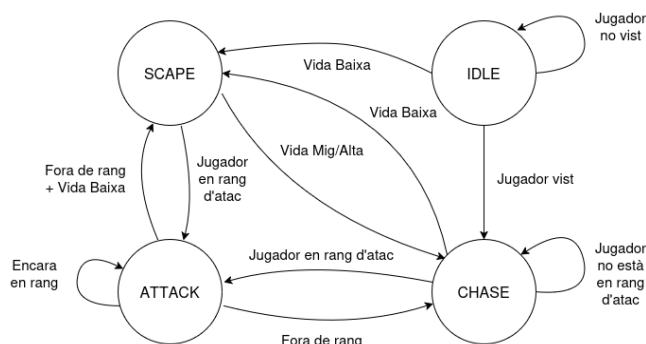


Fig. 10: Diagrama màquina d'estats dels enemics

Cada enemic té una puntuació de vida i quan el jugador els ataca els hi redueix la vida. Els següents enemics, vists en la figura 11, són els que es podran trobar en les sales:



Fig. 11: Enemics. D'esquerra a dreta: Orc, Trol, Parca i Porc Senglar

En la següent taula es reflecteixen les característiques de cada tipus d'enemic:

Enemic	Tipus	Visió	Vida
Orc (Arquer)	A distància	Con poc ample Gran distància	6
Trol (Expadatxí)	Cos a cos	Con mig ample Mitja distància	9
Parca (Bruixot)	A distància	Mínim ample Màxima distància	6
Porc Senglar (Guerrer)	Cos a cos	Màxim ample Mínima distància	12

7.5 Enemic Final

La sala final és l'última sala del joc i es on es troba l'enemic final (figura 13). Pot ser de 3 tipus Foc, Planta o Aigua; aquest tipus es determina a l'inici de cada partida. En la següent figura, la figura 12, veiem els 3 tipus d'enemic final.



Fig. 12: Enemics finals. D'esquerra a dreta: Foc, Aigua i Planta

El enemic final comparteix el model de màquina d'estats com els altres enemics del joc però sense l'estat *Scape* (Com es veu al diagrama de la figura 10 sense *Scape*). Segons la dificultat, la seva quantitat de vida variarà.

Dificultat	Vida
Fàcil	30
Normal	100
Difícil	250

La batalla final s'inicia quan el jugador s'apropa a l'enemic final i la batalla consta de 2 fases. La primera, on l'enemic ataca a distància (Valor del dany 5), i la segona, que comença quan la vida de l'enemic baixa per sota de la meitat de la vida base, on l'enemic ataca a cos a cos (Valor del dany 3). En finalitzar la batalla, s'obre un camí que el jugador travessa marcant el final del joc.



Fig. 13: Captura de la batalla final a la sala final

8 RESULTATS

S'ha obtingut un joc roguelike funcional amb un el menú principal, on es pot començar una partida, sortir del joc o seleccionar una dificultat. El joc compte amb un submenú amb les dificultats disponibles Fàcil, Normal, Difícil i Personalitzada. Un submenú per la dificultat personalitzada que conté:

- Tria de valors de generació: Nombre de sales intermèdies, Nombre de sales font, Nombre d'enemics a les sales, quins enemics estan disponibles en les sales i quins tipus d'enemics finals estan disponibles.
- Atributs enemics: Vida de cadascun, valors d'atac, velocitat, mal extra segons el tipus d'enemic final
- Atributs Jugador: Vida, valor d'atac, velocitat, vida a restaurar a les sales font, valor de millora d'atac, disponibilitat d'atac làser, el mal que fa el làser i els punts necessaris per desbloquejar-lo

En el joc hi ha una sala inicial, diferents sales per les quals travessa el jugador que es generen aleatòriament segons el nombre de sales indicat per la dificultat, 4 enemics bàsics que es poden trobar en aquests (Orc, Trol, Parca i Porc Senglar), una sala font per millorar el personatge (la seva vida, els atacs bàsics o desbloquejar l'atac làser) i la sala final amb un enemic final que pot ser de tipus foc, planta o aigua.

Els resultats obtinguts al final del projecte es poden veure en el vídeo de la figura 14. Aquest mostra una partida de nivell normal on es poden veure la majoria d'elements implementats en el projecte.

Fig. 14: Enllaç del video: <https://youtu.be/F-zguRlvDFg>

9 CONCLUSIONS

Amb els resultats obtinguts es pot dir que s'han complert els objectius plantejats a l'inici del projecte. Utilitzant el game engine Godot s'ha desenvolupat un joc de gènere roguelike completament funcional amb un inici i final establerts. Cada partida és completament diferent ja que s'ha implementat una generació procedural de sales.

Aquest joc compta amb menú de selecció de dificultat i de configuració d'elements del joc (nombre de sales intermitjes, nombre de sales font, nombre i dificultat dels enemics i configuració d'atributs del jugador) i amb enemics modelats amb màquines d'estats.

9.1 Experiència personal

Sent aquesta la primera experiència en desenvolupant un joc més o menys complex i fent-ho amb el motor de Godot, estic satisfet amb els resultats obtinguts, havent pogut respectar en gran part la planificació de les tasques que havia pensat inicialment.

El procés de desenvolupament m'ha permès adquirir noves habilitats, així com enfrontar-me a desafiaments tècnics nous als quals no m'havia enfrontat anteriorment. Aquesta experiència m'ha ajudat a comprendre millor el desenvolupament de videojocs. Estic motivat per continuar aprenent i perfeccionar el treball que he fet.

A més, he pogut introduir-me satisfactòriament al motor Godot. He après quina és l'abstracció que utilitza per construir els jocs, basada en l'ús de nodes ordenats en forma d'arbre per crear escenes que també es poden combinar. Amb aquesta filosofia, he adquirit coneixements sobre com dividir un joc en escenes, com reutilitzar-les i com fer que interactuin entre elles. Això m'ha permès passar de resoldre problemes petits a combinar-les en grans parts complexes que solucionen reptes més importants.

En últim lloc, vull destacar que un altre aspecte clau del projecte ha estat l'oportunitat de millorar la meua capacitat de documentar processos i resultats, així com esforçar-me per planificar correctament les tasques a realitzar i fer-les dintre del temps establert.

9.2 Problemes durant el desenvolupament

Durant el desenvolupament, certes funcionalitats del joc han causat més problemes que d'altres.

Com que no s'han creat els sprites usats en el joc, moltes vegades les spritesheets que obtenia no estaven del tot ben alineades o havia d'utilitzar varies per obtenir l'efecte que volia. Al no tenir experiència amb la manipulació d'imatges, vaig tardar temps a obtenir els sprites finals com volia.

Un altre problema destacable que vaig tenir va ser durant la creació de la màquina d'estats dels enemics, concretament amb l'estat de Chase on l'enemic persegueix al jugador. Això ho ha de fer esquivant els obstacles que hi ha a les sales, el que em va ocasionar més problemes dels esperats, ja que s'havia d'obtenir enemics que els poguessin esquivar.

Finalment, els atacs del jugador i enemics també em van causar bastants problemes ja que em va costar coordinar bé l'atac amb l'impacte als sprites. El primer enfocament que vaig tenir va ser utilitzar senyals de Godot per indicar quan l'atac impactava, però de la forma en què ho vaig

implementar feia que els atacs es veiessin descoordinats i estranys. Repensant la implementació, vaig aconseguir implementar-ho correctament.

9.3 Treball futur

Com a possibles expansions del treball fet, si es tingués més temps es podria desenvolupar més el comportament dels enemics del joc, en lloc de què fossin implementats amb una màquina d'estats es podrien implementar amb intel·ligència artificial perquè no tinguessin uns moviments tan repetitius.

Es podria implementar també un altre tipus de sales opcionals on el jugador s'hagués d'enfrontar a un repte que comportés un risc alt com enfrontar-se a una horda d'enemics que anessin apareixent durant un temps i que, si sobrevisqués, obtingués una millora extra.

També es podria expandir el joc més enllà de l'enemic final passant de la zona de la cova a la zona del bosc que es veu a la sala final, on hi hauria nous enemics i un altre enemic final. Aquesta idea es podria anar escalant tenint diferents zones.

10 AGRAÏMENTS

En primer lloc, vull agrair sincerament a Jorge Bernal, el meu tutor de treball, per la seva implicació, ajuda i consells, que han fet que el treball hagi estat molt més fàcil de portar i l'hagi pogut gaudir més.

I també com no podria faltar, agrair el suport que m'han donat els meus pares i amics no només en el treball sinó durant tota la carrera.

REFERÈNCIES

- [1] Bernice Silva, *What is the difference between rogue-like and RPG?*, 2024, <https://shorturl.at/m92Z1>
- [2] Coleman Gallioreto, *History of the Roguelike, from Rogue to Hades*, 2020, <https://shorturl.at/HS0Dk>
- [3] Álex Pareja, *La emocionante evolución del Roguelike: orígenes, nicho y su llegada a la primera plana en videojuegos como Deathloop, Hades, Returnal...*, 2021, <https://shorturl.at/ezySP>
- [4] Godot <https://godotengine.org/>
- [5] Neilie Johnson, *The Binding of Isaac Review IGN*, 2011, <https://shorturl.at/ZYU7w>
- [6] Hades <https://www.supergiantgames.com/games/hades/>
- [7] Moonlighter <https://moonlighterthegame.com/>
- [8] Roland van der Linden, Ricardo Lopes, i Rafael Bidarra, *Procedural Generation of Dungeons*, 2012, <https://ieeexplore.ieee.org/abstract/document/6661386>
- [9] G. Rozenberg, Ed., *Handbook of Graph Grammars and Computing by Graph Transformation: Volume I. Foundations.*, 1997
- [10] G. Stiny and J. Gips, *Shape grammars and the generative specification of painting and sculpture*, 1971
- [11] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. V. Gool, *Procedural modeling of buildings*, 2006, pp. 614–623
- [12] L. Johnson, G. N. Yannakakis, and J. Togelius, *Cellular automata for real-time generation of infinite cave levels*, 2010, DOI: 10.1145/1814256.1814266
- [13] K. Hartsook, A. Zook, S. Das, and M. Riedl, *Toward supporting stories with procedurally generated game worlds*, 2011, pp. 297–304
- [14] V. Valtchanov and J. A. Brown, *Evolving dungeon crawler levels with relative placement*, 2012, pp. 27–35
- [15] D. Ashlock, C. Lee, and C. McGuinness, *Search-based procedural generation of maze-like levels*, 2011, pp. 260–273
- [16] T. Roden and I. Parberry, *From artistry to automation: A structured methodology for procedural content creation*, 2004, pp. 151–156
- [17] Julia Martins, *¿Qué es la metodología Kanban y cómo funciona?*, 2024, <https://asana.com/es/resources/what-is-kanban>
- [18] Trello <https://trello.com/home>
- [19] Godot Documentation <https://shorturl.at/3yq3G>
- [20] Godot Engine 3.6 documentation in English: Overview of Godot's key concepts <https://shorturl.at/rHWYd>
- [21] ArMM1998, *Zelda-like tilesets and sprites*, 2017, <https://shorturl.at/sITa4>
- [22] GreyFrogGames, *Zelda-like (with oversized sword)*, 2021, <https://shorturl.at/5tjPq>
- [23] Kronovi-, *Boss: Mecha-Stone Golem [FREE]*, <https://shorturl.at/nwv1k>
- [24] Diarandor, *Solarus Fountain*, 2017, <https://shorturl.at/AXGv4>
- [25] Evert, *LPC compatible 3/4 view cave interior*, 2018, <https://shorturl.at/bEAQ3>
- [26] Croatz, *RPG Rocks and Ores*, <https://shorturl.at/F0Gbc>
- [27] Sorthen, *Orcs and Goblins Sprite Pack*, <https://shorturl.at/TXxOm>
- [28] kenam0, *Arrows Pack*, <https://shorturl.at/fFqH5>

- [29] Admurin, *Monster Pack 40*, <https://shorturl.at/2nAf5>
- [30] XYEzawr, *Free Pixel Effects Pack 12 - Mini Magick Shoots 3*, <https://shorturl.at/33KMN>
- [31] unTied Games, *Will's Pixel Explosions*, <https://shorturl.at/EuNtb>
- [32] Free Game Assets (GUI, Sprite, Tilesets), *Free Pixel Predator Plant Mob Sprites*, <https://shorturl.at/wqmzm>
- [33] NYKNCK, *Fireball*, <https://shorturl.at/j8UqK>
- [34] NYKNCK, *Ice Shard Spell Effect*, <https://shorturl.at/xezJI>
- [35] PixiVan, *Simple Forest Tilesset*, <https://shorturl.at/zpS69>
- [36] Disven, *Pixel Swords Assets 16x16*, <https://shorturl.at/RG4oH>
- [37] SirAndriy, *Free icons 16x16*, <https://shorturl.at/iKX2Z>
- [38] BDragon1727, *Basic Pixel GUI and Buttons pack 1* <https://shorturl.at/DliZf>
- [39] Ludenc, *Cave Background and Tilesset* <https://shorturl.at/UTUum>

APÈNDIX

A.1 Diagrama de Gantt

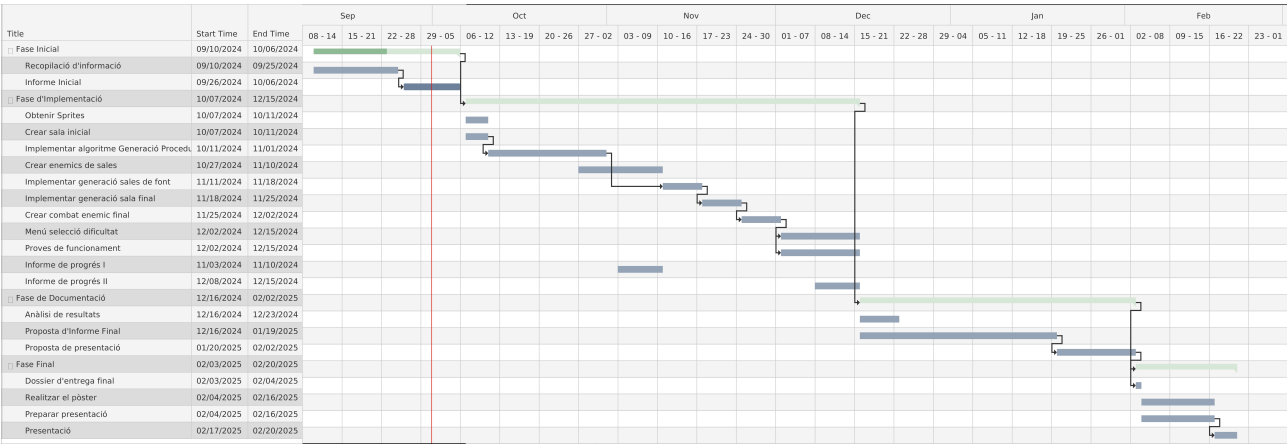


Fig. 15: Diagrama de Gantt del projecte