

Desenvolupament del videojoc Hive Adventure

Daniel Gili López

Grau d'Enginyeria informàtica

Videojocs

Gisela Vaquero Juanola

Joan Arnedo Moreno

Desembre del 2020



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

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Desenvolupament del videojoc Hive Adventure</i>
Nom de l'autor:	<i>Daniel Gili López</i>
Nom del consultor/a:	<i>Gisela Vaquero Juanola</i>
Nom del PRA:	<i>Joan Arnedo Moreno</i>
Data de lliurament (mm/aaaa):	<i>12/2020</i>
Titulació o programa:	<i>Grau d'Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Videojocs</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Videogame, Unity, Hive</i>

Resum del Treball

En aquest treball es dissenya i desenvolupa un videojoc basat en el joc de taula Hive, que és un joc determinista de dos adversaris semblant als escacs. Tenint en compte que el temps i els recursos són limitats, es decideix fer un joc senzill però complet.

El joc disposa de gràfics tridimensionals, efectes sonors, traduccions en tres idiomes i d'un mode aventura amb nivells que donen sensació de progrés al jugador. Es desenvolupa utilitzant l'entorn de Unity i el joc és compatible amb múltiples plataformes.

En el desenvolupament del joc es treballen aspectes com la implementació de les regles, la Intel·ligència artificial, patrons de disseny, estructures de dades i interacció persona ordinador, que són coneixements que s'han adquirit durant el transcurs del grau. A més, s'han adquirit nous coneixements per la creació del joc com l'ús de Unity i conceptes bàsics de Blender.

En la planificació del treball s'ha prioritzat el tenir una versió funcional del joc com més aviat millor per poder fer front als possibles problemes que puguin sorgir. Cal destacar que el joc s'ha desenvolupat amb èxit i s'ha aconseguit realitzar una versió completament funcional.

Abstract

In this work, a game based on the Hive board game is designed and developed, which is a deterministic chess-like game of two opponents. Since time and resources are limited, a simple but complete game is made.

The game features three-dimensional graphics, sound effects, translations in three languages and an adventure mode with levels that give to the player a sense of progress. It is developed using Unity's environment and the game is cross-platform compatible.

In the development of the game, aspects such as rules implementation, artificial intelligence, design patterns, data structures and user experience are worked on, which is knowledge that has been acquired during the degree. In addition, new knowledge has been acquired, such as the use of Unity and basic concepts of Blender.

Work planning prioritized having a functional version of the game as soon as possible to deal with any problems that might arise. Overall, the game has been successfully developed and a full functional version has been made.

Índex

1.	Introducció	1
1.1.	Context i justificació del Treball.....	1
1.2.	Objectius del Treball	1
1.3.	Enfocament i mètode seguit	1
1.4.	Planificació del Treball.....	1
1.5.	Breu sumari de productes obtinguts	4
1.6.	Breu descripció dels altres capítols de la memòria	4
2.	Estat de l'art.....	5
2.1.	Hive	5
2.2.	Gènere	6
2.3.	Joc independent	6
2.4.	Tecnologia.....	6
3.	Definició del joc.....	8
3.1.	Descripció del projecte	8
3.2.	Descripció del joc de taula Hive	8
4.	Disseny tècnic	10
4.1.	Entorn i eines de desenvolupament.....	10
4.1.1.	Unity	10
4.1.2.	Blender	10
4.1.3.	Inkscape	10
4.1.4.	Paint.net	10
4.2.	Recursos	11
4.3.	Estructura de directoris	12
4.4.	Estructura de l'escena	13
4.5.	Estructura del directori Scripts	13
5.	Nucli del joc	16
5.1.	Diagrama de classes UML.....	16
5.2.	Estructures de dades.....	16
5.3.	Tauler hexagonal.....	17
5.4.	Regles	17
5.4.1.	Moviment de les peces	18
5.5.	Intel·ligència Artificial	19
5.6.	Tests unitaris	20
6.	Mode aventura.....	22
6.1.	Objectiu	22

6.2.	Nivells.....	22
6.3.	Editor de nivells	23
6.4.	Emmagatzematge persistent	23
7.	Interacció amb l'usuari	25
7.1.	Menús i navegació.....	25
7.2.	Model de la peça	28
7.3.	Animacions	29
7.4.	Idioma.....	30
7.5.	Efectes de so.....	31
8.	Manual d'usuari	32
8.1.	Requeriments de maquinari.....	32
8.2.	Instruccions del joc	32
9.	Línies de treball futures.....	35
10.	Conclusions	36
11.	Glossari	37
12.	Bibliografia.....	38

Llista de figures

Figura 1. Diagrama de Gantt.....	3
Figura 2. Captura de pantalla de Hive per android.....	5
Figura 3. Captura de pantalla de Hive per PC.....	5
Figura 4. Captura de pantalla de PuyoPuyo Tetris.....	6
Figura 5. Imatge del joc Hive	9
Figura 6. Peça de la reina.....	11
Figura 7 Peça del escarabat	11
Figura 8. Peça de la Llagosta	11
Figura 9. Peça de la formiga	12
Figura 10. Peça de la aranya	12
Figura 11. Diagrama de classes UML	16
Figura 12. Sistema de coordenades hexagonal	17
Figura 13. Captura dels tests unitaris.....	21
Figura 14. Selecció de nivell del mode aventura	22
Figura 15. Editor de nivells.....	23
Figura 16. Diagrama de navegació	25
Figura 17. Pantalla de selecció de mode de joc.....	26
Figura 18. Pantalla de selecció de nivell	26
Figura 19. Pantalla de configuració.....	27
Figura 20. Pantalla de crèdits	27
Figura 21. Pantalla d'una partida	28
Figura 22. Pantalla del menú de pausa.....	28
Figura 23. Model de la peça a Blender	29
Figura 24. Textura d'una peça	29
Figura 25. Animator de Unity	30
Figura 26. Instruccions d'una partida	33

1. Introducció

1.1. Context i justificació del Treball

El present treball s'emmarca dintre del grau Enginyeria Informàtica, concretament dintre de l'itinerari de Enginyeria del Programari. La finalitat del treball és la síntesi i posa en pràctica de tots els coneixements adquirits.

La temàtica del treball ha sorgit de l'afició als jocs de taula i als videojocs i de les ganes de posar en pràctica els algorismes estudiats en l'assignatura de Intel·ligència Artificial.

1.2. Objectius del Treball

- Creació un videojoc totalment funcional.
- Programació de les regles del joc de taula Hive.
- Programació de la intel·ligència artificial a partir dels algorismes estudiats.
- Aprendre a utilitzar eines com Unity, Blender i Git.

1.3. Enfocament i mètode seguit

Es decideix prioritzar la finalització d'un joc complet en tots els aspectes a fer un joc complex. És a dir, s'ha escollit un tema aparentment senzill però que es pot anar ampliant en funció del temps disponible. Per exemple, afegint un mode aventura o altres modes de joc si el temps ho permet.

Per aquest motiu, que s'ha optat per seguir una metodologia àgil en la que s'ha anat dissenyant, programant i provant el projecte al mateix temps, i s'ha prioritzat tenir una versió funcional del joc tant aviat com sigui possible. L'objectiu és poder adaptar l'extensió del projecte al ritme de desenvolupament.

1.4. Planificació del Treball

Aquesta ha estat la planificació que s'ha seguit durant el desenvolupament del projecte. El projecte s'ha dividit en les següents tasques:

- Objectius bàsics:

- Definir l'estructura del projecte: Definir la estructura de fitxers del projecte Unity.
 - Implementar les regles: Consisteix en modelar el tauler i el moviment de totes les peces.
 - Crear els models 3D: Consisteix en crear els hexàgons tridimensionals i afegir les imatges que identifica cada peça.
 - Moviment de les peces en l'espai i interacció amb l'usuari: Definir les animacions de les peces i com hi interactua l'usuari, tenint en compte els moviments que pot fer.
 - Programar la IA: Picar els tres algorismes sobre els que es suporta la Intel·ligència Artificial.
- Menús
 - Especificar els menús: Especificar la funcionalitat i navegació dels diferents menús del joc.
 - Dissenyar els menús: Utilitzant gràfics vectorials si és possible per aconseguir la màxima resolució optimitzant recursos.
 - Animar els menús: Afegir moviment als menús per donar moviment al joc.
 - Programar la funcionalitat dels menús.
- Mode aventura:
 - Definició dels tipus de missions: Definir els tipus de missió i els paràmetres de cada una.
 - Programació dels tipus de missions.
 - Definició de les missions: Crear fitxer JSON o XML amb tota la informació de cada missió: identificador, posició inicial de les peces, tipus de missió, paràmetres, descripció, etc.

- Sistema per guardar el progrés: Consisteix en guardar el progrés de l'usuari en memòria persistent perquè no tingui que tornar a començar de zero cada cop.
- Acabats:
 - Selecció de la música.
 - Efectes de so.
 - Traduccions.
 - Moviment de la càmera: Polir el moviment de la càmera per aconseguir transicions suaus.
- Probes i correcció d'errors.

La distribució en tres lliurables es s'ha realitzat de la següent manera:

- PAC2: Objectius bàsics.
- PAC3: Menús i mode aventura.
- PAC4: Acabats i correcció d'errors.

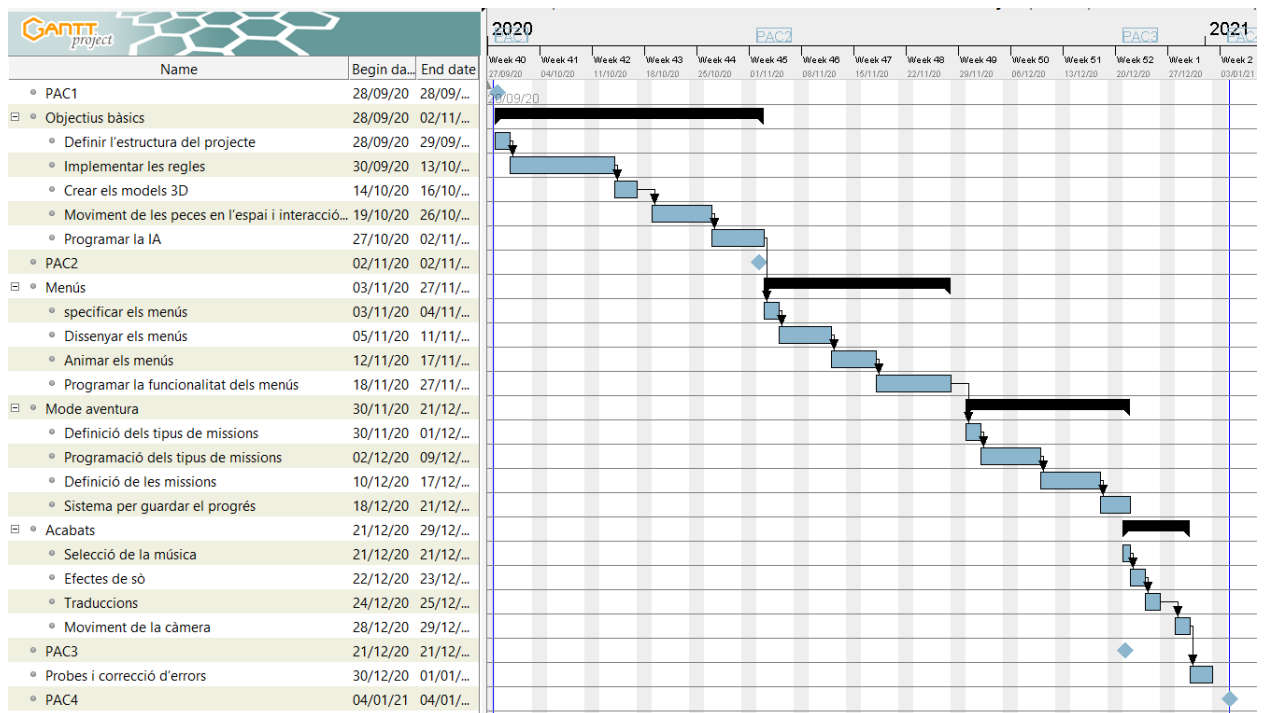


Figura 1. Diagrama de Gantt

1.5. Breu sumari de productes obtinguts

- Executables del joc per les plataformes Windows, Linux i Android.
- Showcase del joc a Unity Connect. <https://connect.unity.com/p/hive-adventure>
- Projecte amb el codi font i el executables a GitHub. <https://github.com/danigili/HiveAdventure>
- Memòria final del projecte.

1.6. Breu descripció dels altres capítols de la memòria

Estat de l'art: en aquest capítol es parla sobre els videojocs basats en el joc de taula Hive, sobre l'estat dels jocs independents i sobre altres jocs similars. També es parla de les eines de desenvolupament que hi ha disponibles avui dia.

Definició del joc: Es tracta d'una breu descripció de l'abast del projecte.

Disseny tècnic: En aquest apartat es descriuen les eines utilitzades i s'explica l'estructura de directoris, estructura de l'escena de Unity i es fa una petita descripció de tots els Scripts que componen el joc.

Nucli del joc: Aquest és el principal capítol del projecte. En ell s'explica la implementació de les regles del joc i de la Intel·ligència Artificial.

Mode aventura: S'explica en que consisteix el mode Aventura, alternativa a la partida simple contra la màquina.

Interacció amb l'usuari: S'expliquen conceptes relatius a l'experiència d'usuari com la navegació dels menús, gràfics, els efectes de so i traduccions entre altres.

Manual d'usuari: Requisits recomanats i manual del joc.

Treballs futurs: Es llisten possibles línies de treball en les que es podria treballar en cas de voler ampliar el videojoc.

Conclusions: Valoració del treball realitzat.

2. Estat de l'art

A continuació es descriu l'estat de l'art tant dels videojocs basats en Hive, com del gènere i de la tecnologia disponible per a desenvolupar videojocs.

2.1. Hive

Aquest no és el primer cop que Hive dona la benvinguda al món dels videojocs. Existeixen nombroses implementacions del conegut joc de taula. Aquesta però, és la primera que incorpora un mode aventura amb diferents reptes per l'usuari.

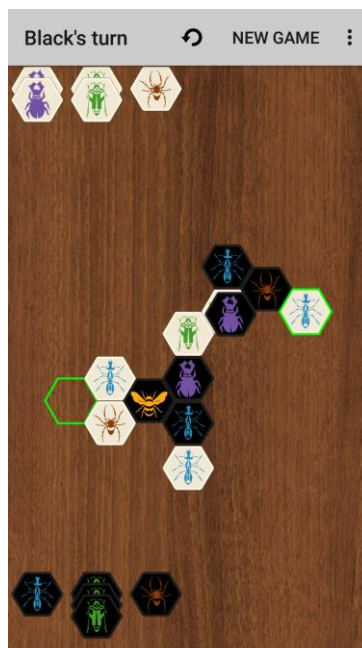


Figura 2. Captura de pantalla de Hive per android



Figura 3. Captura de pantalla de Hive per PC

Existeix videojoc oficial per ordinador, mòbil i navegador web. A més, hi ha diverses versions de codi lliure a GitHub desenvolupades en javascript i python principalment.

2.2. Gènere

Per altre banda, hi ha diversos jocs senzills i coneguts que incorporen un mode aventura amb missions. Per exemple, “PuyoPuyo Tetris” agafa el tradicional Tetris i li afegeix diferents modes de joc i una campanya bastant interessant. Altres exemples de jocs d’aquest estil serien el Candy Crash, Angry Birds o Pang Adventure.



Figura 4. Captura de pantalla de PuyoPuyo Tetris

2.3. Joc independent

Tenim que recordar que aquest es tracta d'un joc desenvolupat per una sola persona amb recursos limitats. En aquest marc s'han desenvolupat molts jocs, alguns d'ells molt coneguts. Alguns exemples són Axiom Verge i Cave Story. Si estudiem aquests jocs, s'observa que la majoria de creadors tenen un perfil principalment artístic i que els jocs destaquen principalment per la jugabilitat i originalitat, i no tant per l'aspecte gràfic, que sol ser modest.

2.4. Tecnologia

Avui dia hi ha molts entorns i eines gratuïtes que permeten realitzar videojocs. Ja no cal desenvolupar el joc des de zero en un llenguatge de programació, sinó que existeixen entorns com Unity, Unreal Engine o GameMaker d'entre altres, que ofereixen totes les eines necessàries.

Aquests entorns integrats ofereixen funcionalitats com un motor gràfic, interacció amb l'usuari, Shaders, scripting, gestió de l'àudio, il·luminació, intel·ligència artificial, etc.

Amb GameMaker es poden fer videojocs senzills sense necessitat de tenir coneixements de programació. Posat que aquest projecte forma part de l'itinerari d'Enginyeria del Programari del grau d'Enginyeria Informàtica, interessa que el projecte tingui un fort component de programació. Per tant, es descarta l'ús de entorns que no requereixen d'aquests coneixements, ja que estan orientats a un perfil més artístic.

Per altre banda, Unreal Engine s'orienta a un perfil més professional i va dirigit a produccions més grans que un projecte de final de grau. Ja que l'entorn és més complex. És per aquest motiu que s'opta per utilitzar Unity, orientat a un perfil més independent, per a jocs més modestos. A més, és fàcil d'utilitzar i ofereix la flexibilitat de poder programar components propis.

3. Definició del joc

3.1. Descripció del projecte

L'objectiu del projecte és implementar el joc de taula Hive. A part d'implementar les regles del joc, es desenvoluparà la IA i es crearà un mode Aventura que enfrontarà al jugador a diferents reptes o missions que ha d'anar superant per tal de completar el joc.

Les missions consistiran en acabar una partida començada amb diferents nivells de dificultat i amb diferents restriccions, com per exemple, en el nombre de moviments. D'aquesta manera, el joc oferirà reptes i sensació de progrés per combatre la sensació d'avorriment.

S'afegiran animacions als diferents elements de la interfície gràfica que aportaran moviment al joc. Així mateix, el joc tindrà un estil *cartoon* amb colors vius i plans, i no es farà ús de textures realistes.

3.2. Descripció del joc de taula Hive

Hive és un joc de taula molt semblant als escacs. Les principals similituds són que el joc és determinista (no hi ha atzar i els dos jugadors veuen totes les peces), cada peça es mou d'una forma diferent i l'objectiu és matar al rei (en aquest cas la reina) que és la peça amb el moviment més reduït. La principal diferència rau en el tauler, en els escacs s'utilitza un tauler quadrat limitat a 8x8 caselles. En canvi, Hive no utilitza cap tauler, si no que les peces es col·loquen sobre una superfície plana, en conseqüència el tauler és il·limitat.



Figura 5. Imatge del joc Hive

Autor: Mike Hulsebus

En el següent enllaç es pot trobar la versió en català de les regles del joc:

<https://lu2.cat/shop/hive-joc-de-taula/>

4. Disseny tècnic

4.1. Entorn i eines de desenvolupament

Tot seguit es llisten les eines utilitzades en el desenvolupament del projecte.

4.1.1. Unity

Unity és un potent motor de videojocs multiplataforma i amb gràfics 3D. És un programa propietari, però disposa de llicències gratuïtes personals i per estudiants que permeten accedir a tota la funcionalitat del motor. Per consegüent, s'ha format una gran comunitat de desenvolupadors al voltant de Unity.

El motor disposa d'eines per representar gràfics en tres dimensions, disposa també d'un editor d'animacions, gestió d'efectes de so, sistema de físiques realistes, sistema de UI per crear els menús, etc. En definitiva, conté totes les eines necessàries per crear un videojoc. És per aquest motiu, que s'ha escollit Unity per a implementar el projecte.

4.1.2. Blender

Per a modelar el disseny de les peces s'utilitzarà Blender, que és un programa de modelatge de codi lliure molt popular.

4.1.3. Inkscape

Inkscape és un editor de gràfics vectorials de codi lliure. Permet generar fitxers SVG (Scalable Vector Graphic) que no perden qualitat al ampliar-los. S'utilitzarà per a crear les imatges de la interfície gràfica.

4.1.4. Paint.net

Paint.net és un programa gratuït d'edició d'imatges. S'ha escollit aquest programa perquè és molt simple i senzill d'utilitzar i permet crear imatges amb transparències i disposa de gestió de capes.

4.2. Recursos

Tots els recursos que s'han utilitzat per a fer el joc són originals excepte les imatges del insectes. Que s'han extret de pàgines d'icones vectorials sense drets d'autor o amb llicència d'ús no comercial. A continuació es fa un recull de les imatges utilitzades.

Reina



Figura 6. Peça de la reina

Escarabat



Figura 7 Peça del escarabat

Llagosta

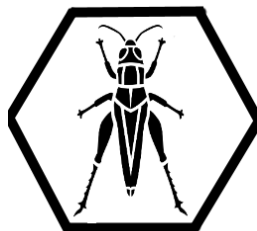


Figura 8. Peça de la Llagosta

Formiga



Figura 9. Peça de la formiga

Aranya

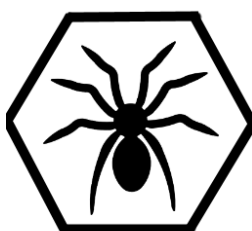


Figura 10. Peça de la aranya

4.3. Estructura de directoris

Els recursos i el codi del joc s'organitza en directoris seguint la següent estructura. S'han organitzat de la forma més ordenada possible:

- GUI: Aquest directori conté tots els elements de la interfície d'usuari. En particular, inclou imatges vectorials en format SVG fetes amb InkScape, fonts, animacions i prefabs.
- Models: conté el model 3D de l'hexàgon.
- Resources: conté elements que Unity carrega en temps d'execució des dels scripts amb la comanda Resource.Load. Esta dividit en diferents directoris:
 - Images: conté les imatges de les peces que encara no estan en joc.
 - Textures: conté les imatges de les peces que estan en joc.
 - Adventure: conté fitxers en format JSON amb tots els nivells del mode aventura organitzats per zona.
 - Translate: conté un fitxer de traducció per a cada idioma.
- Scenes: conté les escenes del joc.
- Scripts: conté tot el codi font.
- SFX: conté els efectes de so.

- Test: conté els tests unitaris de les regles i de la IA.

4.4. Estructura de l'escena

Aquesta és l'estructura de l'escena principal del joc.

- Main Camera: Representa la única càmera del joc
- Directional Light: Conté les propietats de la il·luminació: direcció de la llum, intensitat i propietats de les ombres.
- GameMain: Conté la gestió dels menús i la partida en curs. Conté dos pools d'objectes, un pels marcadors i un altre per les peces.
- Canvas: Conté tots els elements de la interfície d'usuari bidimensionals. Està dividit en diferents panells:
 - Game: Conté les peces pendents de col·locar.
 - GUI: Conté els menús Start, Pause menú, end game.
- IntegratedUI: Conté els elements de la part de la interfície d'usuari que està integrada en el joc, i que per tant, són tridimensionals.

4.5. Estructura del directori Scripts

En el entorn Unity, els pedaços de codi que complementen els components de Unity escrits per el desenvolupador s'anomenen Scripts. Aquests scripts s'han organitzat en directoris en funció de la seva funcionalitat. A continuació es llisten tots els Scripts que componen el joc juntament amb una breu descripció.

- Model: Aquest directori conté els scripts que conformen el nucli del joc. És a dir, la implementació de les regles i la Intel·ligència Artificial.
 - AI: Implementació de l'algorisme minimax amb poda alpha-beta.
 - Board: Conté les regles del joc.
 - Move: Representa el moviment d'una peça a una posició lliure.
 - Piece: Conté la informació d'una peça (posició, tipus i color)
 - Position: Representa una posició en coordenades hexagonals.
 - Serializer: Permet serialitzar i deserialitzar classes a format JSON.

- Game: Conté els scripts que controlen la interacció del jugador amb el tauler.
 - BoardView: Classe principal que controla la interacció entre el jugador i la classe Board.
 - CameraController: S'encarrega de controlar la posició de la càmera.
 - HexObject: Classe abstracta que converteix les coordenades hexagonals al sistema de coordenades cartesià.
 - Marker: Representa un marcador en l'espai del joc.
 - PiecePosition: Representa una peça en l'espai del joc.
 - Pool: Classe que permet reutilitzar objectes com peces o marcadors.
- Adventure: Aquest directori conté els scripts que conformen el mode de joc aventura.
 - AdventureButton: Controla el moviment i les accions dels botons de selecció de nivell.
 - AdventureMenu: Controla el menú de selecció de nivell del mode aventura.
 - SaveGame: Classe que s'encarrega de guardar la partida en emmagatzematge persistent.
- UI: Conté scripts que complementen la interfície d'usuari.
 - IntegratedButton: Controla el moviment i les accions dels botons selectors del mode de joc.
 - Localization: S'encarrega de carregar els fitxers de traducció.
 - TextUI: Classe que tradueix els textos de la interfície gràfica.
 - PiecesPanel: Gestiona els panells laterals que contenen les peces pendents de ser jugades.
- Settings: Scripts del menú de configuració.

- DisableOnAndroid: Desactiva un objecte en l'entorn Android. S'utilitza per desactivar la opció de pantalla completa en dispositius mòbils.
- FullScreenSelector: Controla l'interruptor de pantalla completa.
- LanguageSelector: Controla l'interruptor de selecció d'idioma.
- Volume: Controla la barra de selecció del volum del joc.
- GameMain: És el script principal i gestiona aspectes com la navegació entre les diferents pantalles i l'inici i finalització de les partides.
- LevelEditor: Controla l'escena LevelEditor i permet fer accions com moure, crear o eliminar peces del tauler, així com carregar i guardar jugades en format JSON.

5. Nucli del joc

El nucli del joc o model està format per la implementació de les regles i la intel·ligència artificial. Implementar les regles del joc és la part del projecte més complexa. Ja que Hive, a diferència d'altres jocs com els escacs, no utilitza un tauler. En conseqüència, no es pot utilitzar una matriu per emmagatzemar la posició de les peces. A més, algunes de les regles són complexes d'implementar. En aquest apartat s'explicarà en llenguatge natural la implementació de les regles i les estructures de dades utilitzades.

5.1. Diagrama de classes UML

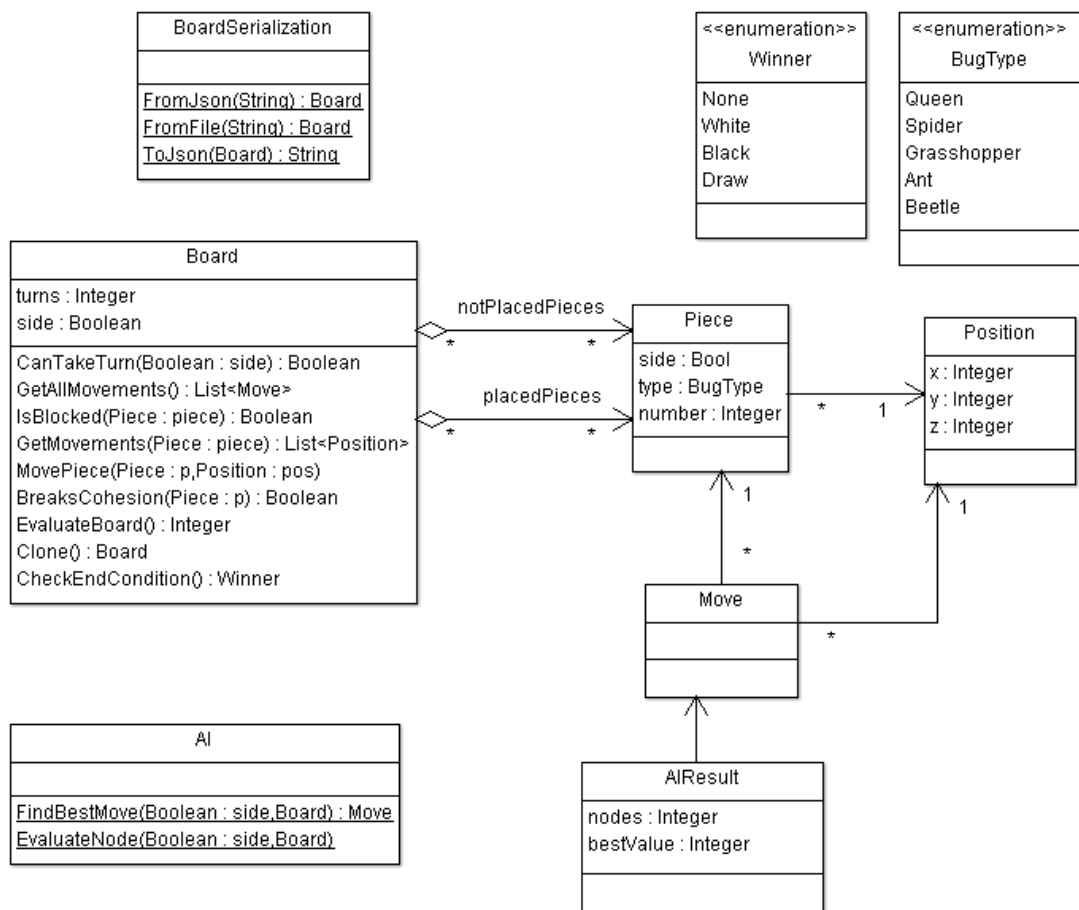


Figura 11. Diagrama de classes UML

5.2. Estructures de dades

S'han utilitzat dues estructures de dades:

- Una llista encadenada per emmagatzemar les peces que encara no han estat jugades.
- Un diccionari per emmagatzemar les peces col·locades al tauler. S'ha utilitzat la posició com a clau. S'utilitza un diccionari ja que el tauler no té límits (i per tant, no es pot utilitzar una matriu) i el número de peces no és fix. S'utilitza la posició com a clau ja que es realitzen moltes consultes per posició i el cost computacional d'accés al diccionari és constant.

5.3. Tauler hexagonal

El tauler és hexagonal, això significa que cada peça és adjacent a unes altres sis peces. La funció GetPosition del model calcula les sis adjacències donades unes coordenades. El vector que retorna representa les sis posicions en ordre horari començant per l'adjacència superior.

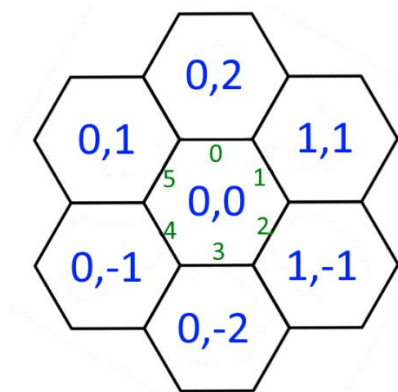


Figura 12. Sistema de coordenades hexagonal

5.4. Regles

Posar una peça en joc. Per jugar una peça, s'ha de col·locar adjacent a qualsevol de les peces però no es pot col·locar adjacent a una peça de l'adversari. A més, s'ha de col·locar lliscant des de fora del tauler. Per aquest motiu, s'utilitza la funció del moviment de la formiga per calcular el perímetre i a continuació, s'eliminen les posicions que són adjacents a peces de l'adversari.

Només hi pot haver un eixam. Aquesta regla impedeix que es pugui moure una peça si divideix l'eixam en dos. Per calcular si una peça trenca aquesta regla i per tant, no es pot moure, s'aplica el següent algorisme.

```
S'elimina la peça que es vol moure
Es selecciona una peça qualsevol
```


Es col·loca la peça en un set sense repeticions

Per cada peça del set

S'afegeixen les peces adjacents (veïnes) al set

Si la mida del set == nombre de peces en joc - 1 - nombre escarabats apilats

la peça es pot moure.

En cas contrari

La peça no es pot moure

Fi de la partida. La partida acaba quan una de les reines o les dues estan rodejades completament. De manera que, si una de les reines té sis peces adjacents, el jugador d'aquesta reina perd. Tanmateix, si les dues reines estan completament rodejades, es considera empat.

5.4.1. Moviment de les peces

La **reina** es pot moure una posició lliscant. Això vol dir que no es pot moure a una posició en la que no hi cap. A continuació es descriu l'algorisme utilitzat:

S'inicialitza la llista de moviments

Per cada una de les sis posicions adjacents a la reina

Si a la posició anterior o següent a la actual no hi ha cap veí, la reina no es pot moure a aquesta posició perquè quedaria flotant.

Si a la posició anterior i a la posició següent hi ha veïns, no es pot moure perquè no hi cap.

En cas contrari, s'afegeix la posició actual a la llista de moviments.

El moviment de l'**aranya** és semblant al de la reina però la peça s'ha de moure tres posicions en comptes d'una. De manera que, es reutilitza el moviment de la reina i s'executa tres cops en cada direcció.

La **formiga** es pot moure lliscant a qualsevol direcció. També, es reutilitza el moviment de la reina i s'executa fins que la formiga dona tota la volta al rusc.

El moviment de la **llagosta** és el més senzill de tots. Els moviments es calculen de la següent manera.

Per cada veí de la llagosta

Posició = posició del veí

```
Mentre hi hagi un veí a posició
    Posició = posició en la mateixa direcció
Afegim la posició a la llista de moviments.
```

L'**Escarabat** només es pot moure una posició, però a diferència de la reina, l'escarabat es pot apilar a sobre d'altres peces. Per tant, s'utilitza un algorisme semblant al de la reina però s'afegeixen nous moviments.

5.5. Intel·ligència Artificial

Per a fer l'algorisme que controla l'adversari s'utilitza un arbre de cerca amb adversari per dos jugadors. L'arbre de cerca es sosté sobre tres pilars o algorismes:

Algorisme Minimax: L'algorisme minimax busca dins de l'arbre de cerca la millor jugada tenint en compte que l'adversari sempre escollirà la jugada més favorable. Com que calcular l'arbre complet és computacionalment impossible, es limita la profunditat i s'utilitza la funció objectiu per determinar el valor de cada jugada.

Poda alpha-beta: La poda alpha-beta permet optimitzar l'algorisme minimax fins a un 50%. Això s'aconsegueix eliminant nodes de l'arbre que no cal expandir perquè ja sabem del cert que mai seran seleccionats.

Funció objectiu: Funció que calcula quin jugador està guanyant la partida en funció de la posició de les peces en un moment donat. Dona un valor numèric, que és negatiu si va guanyant el jugador MIN o positiu i va guanyant el jugador MAX.

El mètode EvaluateBoard conté la funció heurística que retorna el valor de la jugada en curs. El valor és positiu si està guanyant el jugador negre i negatiu si està guanyant el jugador blanc. L'algorisme que utilitza per calcular el valor és molt senzill ja que s'ha d'executar milers de cops. És el següent:

```
Valor = 0
Si la reina blanca està en joc
    Si el nombre de veïns és igual a 6
        Valor = 10000
    En cas contrari
        Valor = valor + peces adjacents de la reina blanca * 10
Si la reina no es pot moure
    Valor = valor - 5
Si al reina negre està en joc
```

```
Si el nombre de veïns és igual a 6
    Valor = -10000
En cas contrari
    Valor = valor - peces adjacents de la reina negra * 10
Si la reina no es pot moure
    Valor = valor + 5
Retorna valor
```

Un cop s'ha implementat la funció heurística, s'utilitza l'algorisme minimax i la poda alpha-beta per a calcular tots els moviments possibles fins a tres nivells de profunditat. En altres paraules, l'algorisme pot preveure fins a tres moviments, tardant una mitja d'un segon en jugar el seu torn. Pot ser que l'algorisme trobi jugades amb el mateix valor, en aquest cas, escull una a l'atzar, amb l'objectiu d'afegir un element aleatori que eviti repeticions en les jugades, a fi que el jugador no guanyi sempre fent la mateixa jugada.

Si es detecten victòries en dos punts diferents de l'arbre de cerca, s'escull la jugada més immediata, ja que en cas contrari es podria donar el cas en que l'adversari es quedi en bucle i no arribi a guanyar mai.

5.6. Tests unitaris

S'han creat tests unitaris de la implementació de les regles i de la Intel·ligència Artificial amb l'objectiu de provar aquests components i evitar que futurs canvis o millores del codi provoquin un mal funcionament inesperat.

Els tests unitaris s'han creat utilitzant l'eina "Test Runner" de Unity en mode edició.

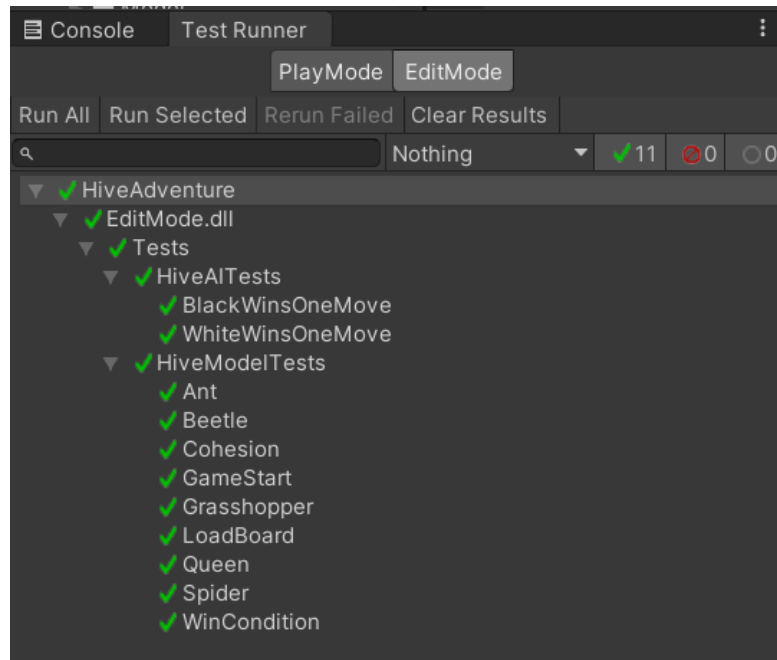


Figura 13. Captura dels tests unitaris

S'han fet test de tots el moviments i de les condicions de finalització d'una partida. A més, es prova que la IA és capaç de acabar una partida que pot guanyar en un sol moviment i que és capaç de bloquejar peces de l'adversari.

6. Mode aventura

6.1. Objectiu

Un bon joc, a part de ser entretingut, ha d'oferir una sensació de progrés al jugador. Amb aquest objectiu en ment, es crea un mode aventura amb nivells que el jugador ha d'anar desbloquejant. Els nivells consisteixen en superar una jugada començada amb un límit de moviments.

6.2. Nivells

Els nivells s'organitzen en zones de 6 nivells cadascuna. Per la primera versió del joc només s'han creat 12 nivells, és a dir, dues zones. La pantalla de selecció de nivell representa els nivells a sobre del tauler de joc. La disposició de les peces en el tauler es defineix en un fitxer JSON, que conté la llista de tots els nivells, i per cadascun, la seva posició en la pantalla. De la mateixa manera, la informació de cada nivell s'emmagatzema en fitxers JSON.

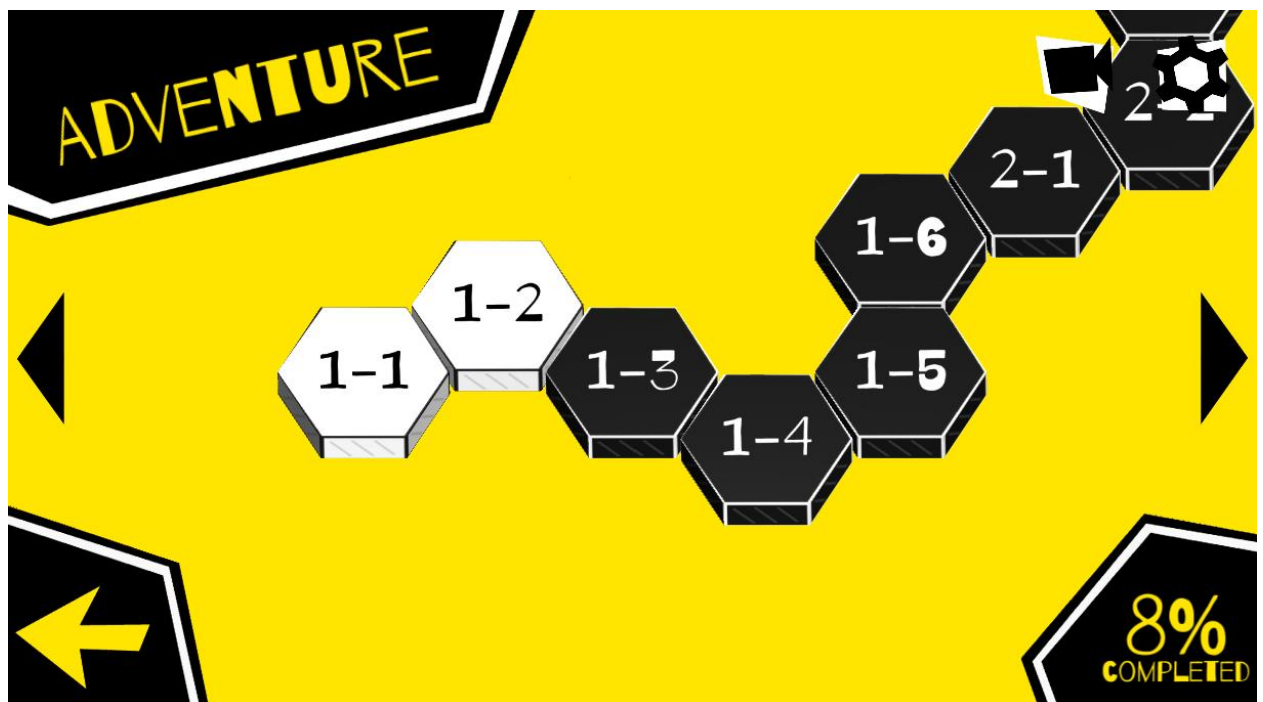


Figura 14. Selecció de nivell del mode aventura

6.3. Editor de nivells

Per facilitar la tasca de dissenyar els nivells al desenvolupador. Es crea un editor de nivells que permet col·locar les peces en el taulell i guardar la jugada en format JSON. A continuació es mostra una imatge de l'editor de nivells.



Figura 15. Editor de nivells

L'editor permet col·locar, crear i eliminar peces. A més, permet indicar el nombre de torns que disposa l'usuari per acabar el nivell.

Cal esmentar que aquesta escena no s'ha depurat ja que no estarà disponible de cara a l'usuari. Per aquest fet, tampoc s'ha treballat l'aspecte gràfic.

6.4. Emmagatzematge persistent

El progrés del joc s'emmagatzema en un fitxer en format JSON. El fitxer només conté una llista dels nivells superats. D'aquesta manera, el fitxer de guardat sempre serà compatible encara que s'afegeixin nous nivells.

La classe SaveGame segueix el patró Singleton i s'encarrega de la gestió del fitxer de guardat. La classe conté els següents mètodes:

- GetInstance: Retorna la instància de la classe SaveGame
- Load: Carrega la partida a partir del fitxer JSON
- Save: Guarda la partida en el fitxer

- Reset: Elimina el fitxer
- LevelCompleted(zone, level): Marca el nivell indicat com a completat i guarda la partida
- IsLevelCompleted(zone, level): Indica si un nivell ha estat completat.

7. Interacció amb l'usuari

7.1. Menús i navegació

En aquest apartat s'explica els menús i la navegació entre les diferents pantalles. El següent diagrama mostra un resum de la navegació entre els menús del joc.

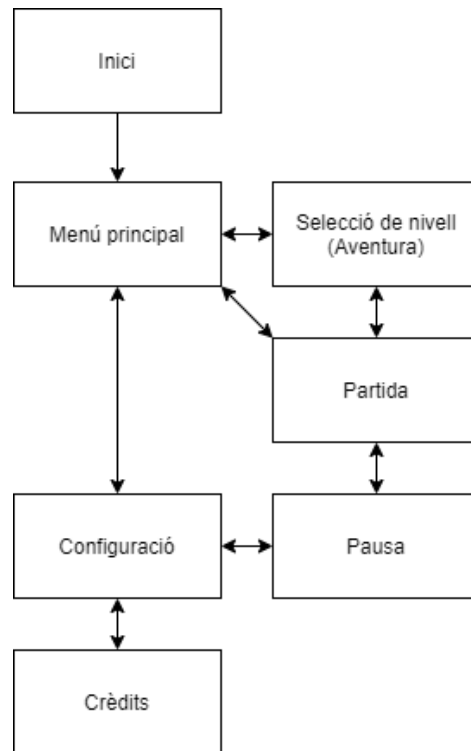


Figura 16. Diagrama de navegació

Al iniciar el joc es mostra el menú d'inici o presentació. A continuació es mostra el menú principal o de selecció de mode de joc, que ens permet escollir un dels tres modes de joc: Partida ràpida contra la IA, partida per a dos jugadors i mode aventura.



Figura 17. Pantalla de selecció de mode de joc

Si es selecciona el mode aventura es carrega el menú de selecció de nivell. Aquesta pantalla permet visualitzar quins nivells s'han superat, el progrés en forma de percentatge, i permet seleccionar el nivell que es vol jugar.



Figura 18. Pantalla de selecció de nivell

Els botons dels dos menús anteriors estan renderitzats utilitzant peces del joc. Per tant, es pot modificar l'angle de la càmera amb el botó en forma de càmera. El botó en forma d'engranatge obre el menú de configuració del joc.



Figura 19. Pantalla de configuració

El menú de configuració permet escollir l'idioma entre català, castellà i anglès, el volum dels efectes de so i permet visualitzar els crèdits. A més a més, en la versió per a ordinador, es pot configurar la pantalla complerta.

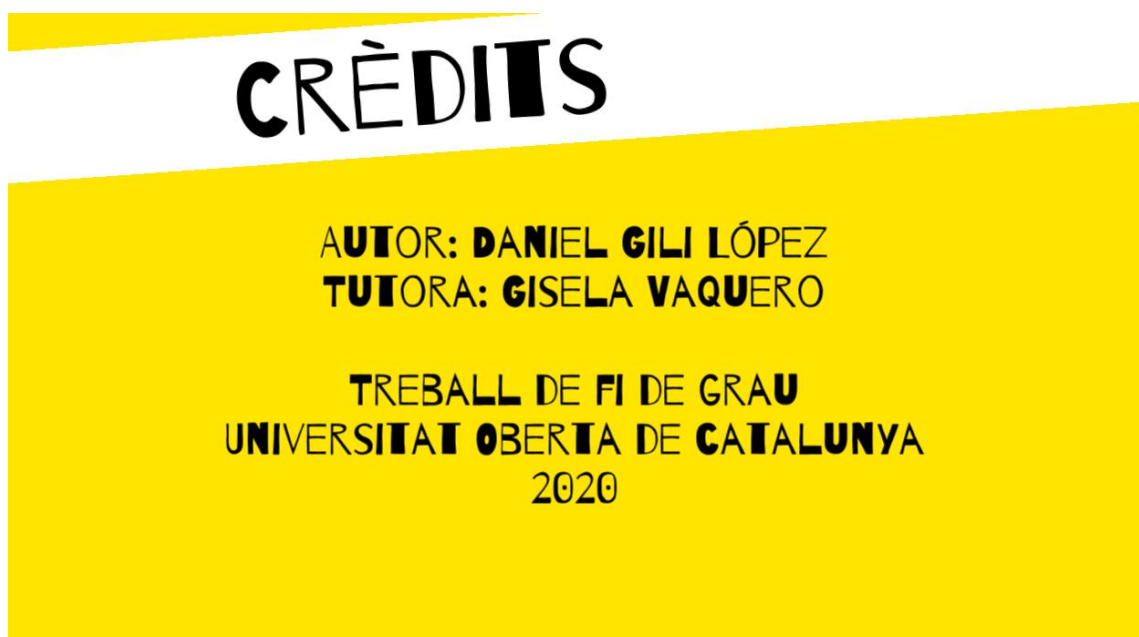


Figura 20. Pantalla de crèdits

Finalment, la pantalla de joc disposa de dos panells laterals amb les peces per jugar de cada participant, un botó per canviar l'angle de la càmera, i el botó que desplega el menú de pausa del joc.

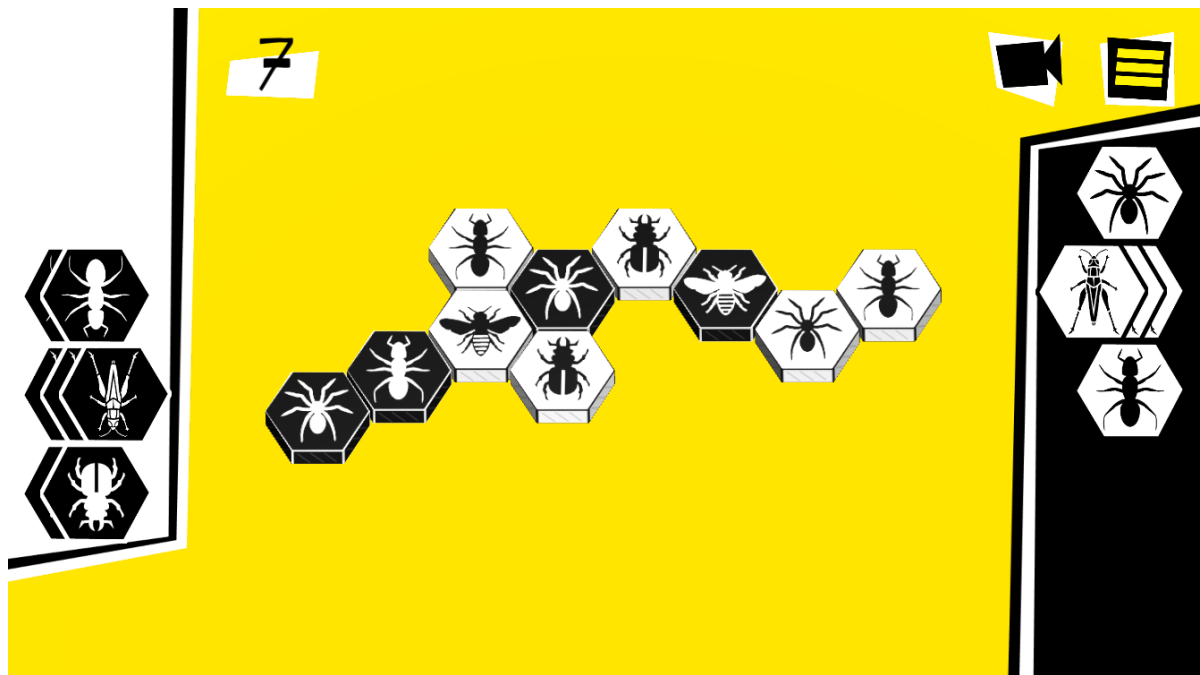


Figura 21. Pantalla d'una partida

El menú de pausa permet reiniciar la partida, obrir el menú de configuració i tornar a la pantalla principal.

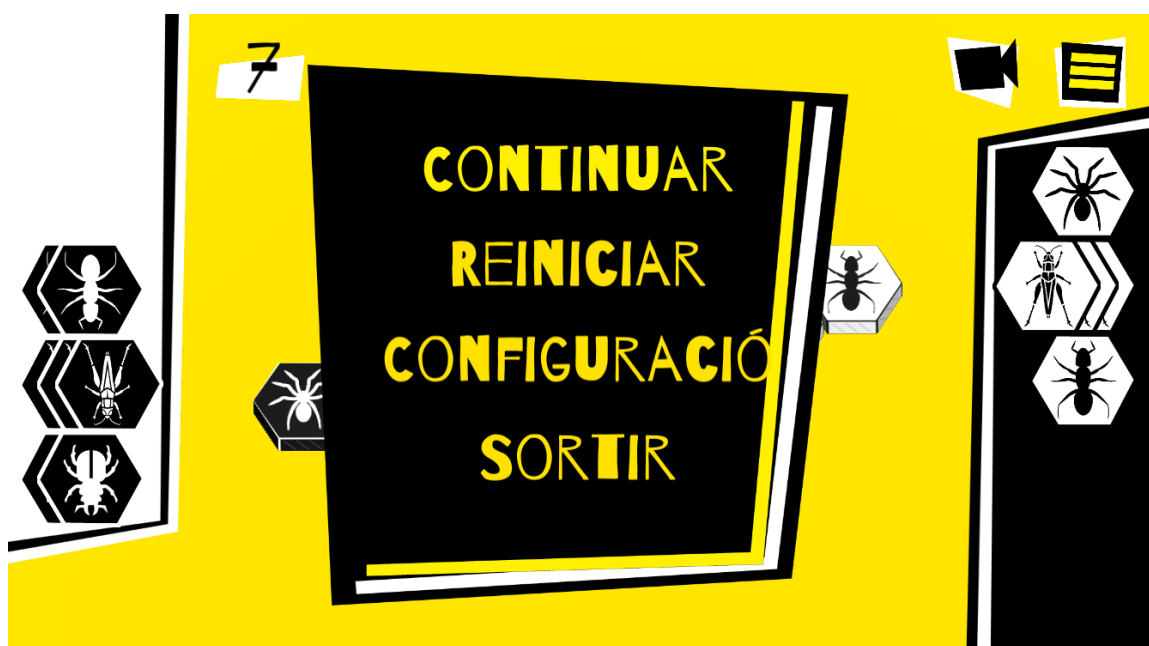


Figura 22. Pantalla del menú de pausa

7.2. Model de la peça

La peça hexagonal s'ha modelat amb Blender i se li ha afegit una textura. La peça s'ha generat extruint un hexàgon. Les diferents peces del joc utilitzen el mateix model i les textures es carreguen des del codi.

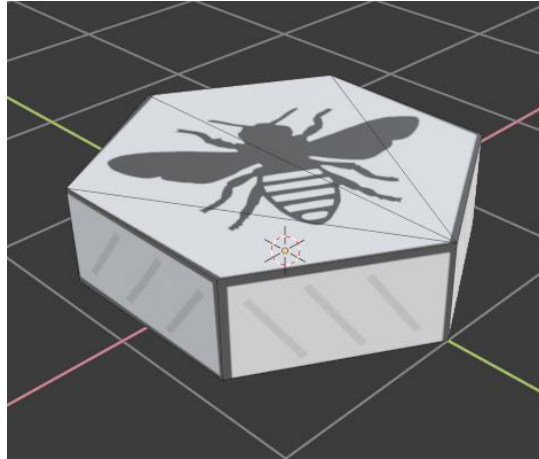


Figura 23. Model de la peça a Blender

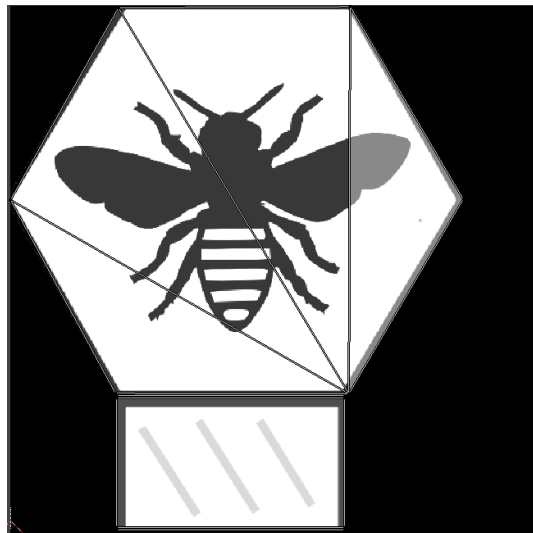


Figura 24. Textura d'una peça

7.3. Animacions

Els menús s'han animat utilitzant l'eina Animator de Unity. Per a cada menú s'ha creat un *AnimationController* amb les fases *hidden*, *show*, *idle* i *hide* i el paràmetre *show* que indica si el menú s'està mostrant en pantalla. La modificació del paràmetre *show* activa l'animació per mostrar o amagar el menú.

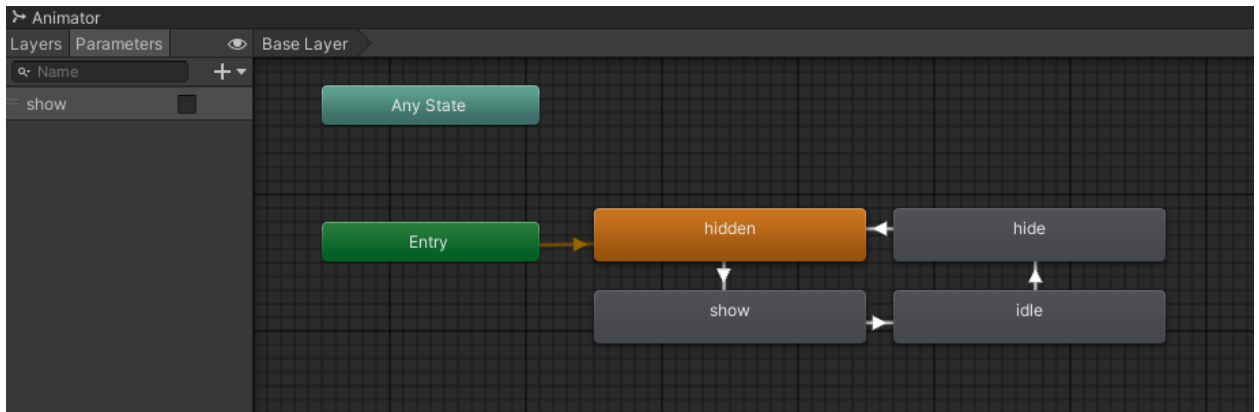


Figura 25. Animator de Unity

Animant els menús es pretén donar una sensació de moviment a un joc que de per si és estàtic.

7.4. Idioma

El joc està disponible en tres idiomes. Perquè la tasca de traducció sigui senzilla, s'ha creat un fitxer de traducció per a cada idioma. Els fitxers de traducció tenen el següent format:

```
WHITE_WINS=Blanques\nguanyen
BLACK_WINS=Negres\nguanyen
DRAW=Empat
ADVENTURE_MODE=Aventura
QUICK_GAME_MODE=Partida\nRàpida
```

En el joc s'utilitza la clau, i la classe Localization s'encarrega de llegir el fitxer corresponent a l'idioma seleccionat i proporcionar el text traduït.

La classe Localization disposa dels següents mètodes públics:

- SetLanguage(Language)
- GetCurrentLanguage(): Language
- Translate(String): String

S'utilitza l'eina PlayerPrefs de Unity per guardar en emmagatzematge persistent l'idioma seleccionat amb l'objectiu de mantenir la selecció de l'usuari entre execucions del joc.

7.5. Efectes de so

Hi ha dos tipus d'efectes de so, el so de col·locar una peça i el so de començar una partida. Els efectes s'han gravat fent xocar les peces del joc entre elles i enregistrat el so amb un micròfon. Val la pena dir que cada so s'ha enregistrat múltiples vegades. D'aquesta manera, s'evita que la repetició del mateix efecte de so cansi al jugador.

8. Manual d'usuari

8.1. Requeriments de maquinari

Els requeriments de maquinari recomanats per executar el joc en PC són:

- Sistema operatiu: Windows 7 SP1+ o Ubuntu 16.04+
- CPU: arquitectura de x64
- GPU: OpenGL 3.2+ o DX10+
- Memòria: 1GB de memòria RAM
- Disc: 100 MB d'espai lliure

Els requisits per a mòbil són:

- Sistema operatiu: Android 4.4+
- CPU: ARMv7 o ARM64
- GPU: OpenGL ES 2.0+, OpenGL ES 3.0+
- Memòria: 1GB+ RAM
- Disc: 100 MB d'espai lliure

8.2. Instruccions del joc

A continuació es donen unes instruccions bàsiques per poder jugar al joc. Primer, cal tenir en compte que la interacció amb el joc es realitza amb el ratolí o amb el dit en dispositius tàctils. Segon, el jugador sempre controla les peces blanques, mentre que la màquina controla les negres.

Hi ha tres modes de joc:

- Partida ràpida: Al seleccionar aquest mode comença una partida contra la màquina.
- Dos jugadors: Per jugar una partida entre dos jugadors.

- Aventura: Aquesta opció carrega el menú de selecció de nivell.

En la següent imatge es mostren els elements d'una partida. Al marge superior dret tenim un botó per alternar entre les tres vistes i tenim un altre botó per obrir el menú d'opcions. A més, hi ha dos panells laterals amb les peces que encara no estan en joc. Finalment, en el mode aventura, apareix un comptador amb el nombre de moviments restants en el marge superior esquerra.

Per moure una peça, només cal seleccionar-la del panell lateral o de les peces del joc (al centre) i es marcarà de color verd les posicions a les que es pot moure la peça. Clicant sobre la posició es mourà la peça. Per altre banda, si es vol cancel·lar el moviment només cal clicar sobre la mateixa peça, clicar-ne una altre o clicar fora del tauler.

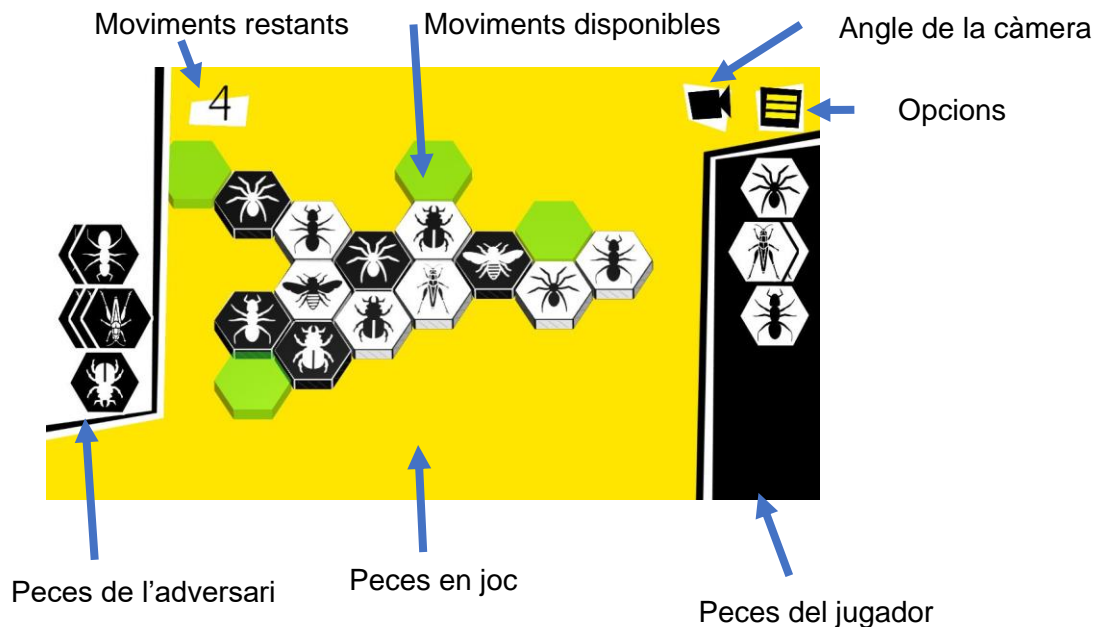
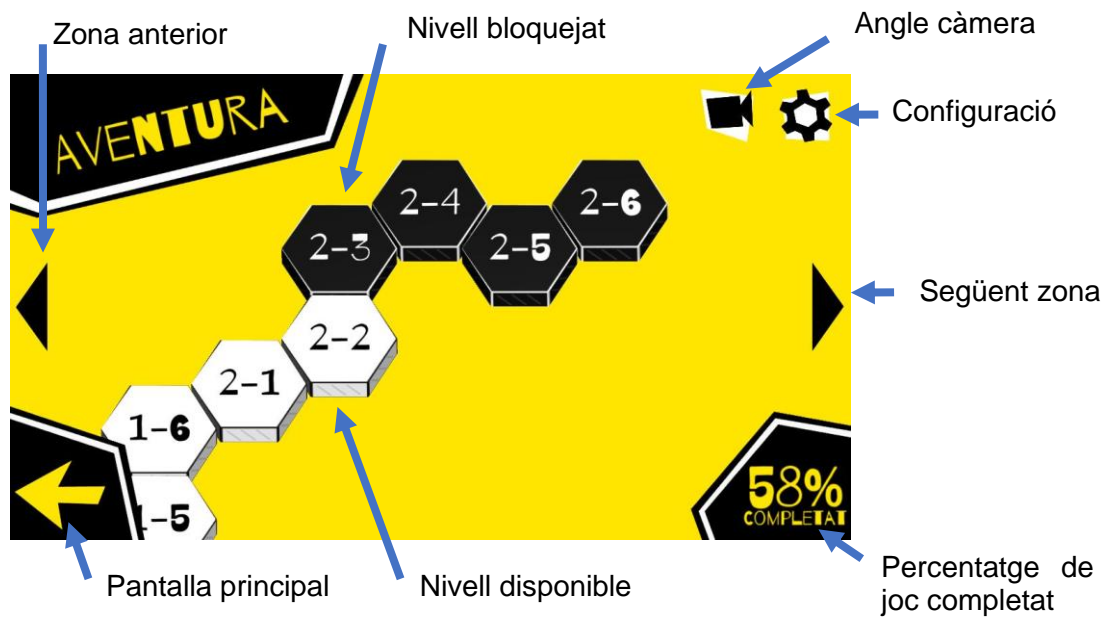


Figura 26. Instruccions d'una partida

En la següent imatge es poden observar els elements de la pantalla de selecció de nivell del mode aventura.



Les fitxes blanques representen nivells que estan disponibles i les negres representen nivells bloquejats. Cada cop que es completa un nivell, es desbloqueja el següent fins a completar tot el joc.

Els nivells consisteixen en guanyar una partida començada amb un nombre de moviments determinats. Si es perd la partida o s'acaba el nombre de moviments i no s'ha guanyat la partida, el nivell es considera no superat i es reinicia. En cas contrari, es torna al menú de selecció de nivell i es desbloqueja el següent nivell.

Les regles del joc de taula es poden trobar a la web oficial.

9. Línies de treball futures

En aquest apartat es descriuen aspectes del joc que es poden ampliar en futures línies de treball.

- **Multijugador en línia:** Consisteix en afegir un mode de joc per jugar amb un adversari en línia, desenvolupant un sistema de emparellament en funció del nivell del jugador.
- **Nous tipus de missions:** Com que el projecte està molt limitat en quant a temps de desenvolupament, només s'ha implementat un tipus de missió i uns pocs nivells. Aquest aspecte del joc té molt marge de millora. Incloure nous tipus de missions o noves mecàniques permetria augmentar el número de missions, augmentant la duració del joc.
- **Música:** Tot i que el joc té efectes de so, no s'ha treballat l'aspecte musical. Un parell de cançons, una durant la partida i una altre en els menús principals donaria més dinamisme al joc.
- **Compatibilitat amb teclat i gamepad:** Actualment l'única forma d'interactuar amb el joc és amb el ratolí o pantalla tàctil. La compatibilitat amb gamepad oferiria la possibilitat de poder exportar el joc a altres plataformes, i donaria més opcions a l'usuari per escollir la forma d'interacció preferida.
- **Expansions:** El joc de taula Hive es complementa amb tres expansions oficials. Cada expansió afegeix una nova peça amb un moviment nou. A més, existeixen altres expansions no oficials. Una possible millora seria implementar totes les expansions.
- **Estrelles:** Consisteix en afegir tres objectius per nivell i representar amb una estrella cada un dels objectius. Aquest sistema l'utilitzen molts jocs com per exemple l'Angry Birds.
- **Nivell de dificultat:** Consisteix en millorar la IA i oferir al jugador la opció de escollir el nivell de dificultat.

10. Conclusions

El joc s'ha desenvolupat amb èxit i s'ha aconseguit fer una versió completa i funcional del programa, de la intel·ligència artificial i del mode aventura. Tot i això, hi ha alguns aspectes que no s'han treballat gaire per falta de temps. Per exemple, la intel·ligència artificial no és suficientment completa com per poder establir diferents nivells de dificultat. A més, només s'han creat 12 nivells pel mode aventura.

La planificació inicial s'ha seguit correctament i s'han assolit tots els objectius plantejats a temps. Això és degut a que s'havia fet una planificació realista amb marge suficient per resoldre els imprevistos.

En total s'han escrit unes 3000 línies de codi, repartides en 30 fitxers i s'han realitzat un total de 96 *commits* al Repositori git. A més, el projecte està format per uns 250 fitxers entre imatges, nivells, fitxers de traducció, animacions, etc.

En el transcurs del treball s'han aplicat coneixements adquirits durant el curs. Concretament de Intel·ligència artificial, patrons de disseny i Interacció persona ordinador entre d'altres. A més, s'han adquirit nous coneixements sobre Unity i Blender.

Com a línies de treball futures hi ha, fer probes amb usuaris per testejar el joc i depurar els possibles errors que pugui tenir, desenvolupar més el mode aventura amb més tipus de nivells i la creació d'un mode multijugador en línia.

11. Glossari

- Singleton: Patró de disseny que només permet una sola instància de la classe.
- Test unitari: Test automàtic que comprova el correcte funcionament d'una part del programa de forma aïllada de la resta.
- Escena: Element de Unity que agrupa tots els elements del joc de forma jeràrquica.
- Script: En entorn Unity són trossos de codi escrits pel desenvolupador que complementen els components de Unity.
- SFX: acrònim en anglès de efectes sonors (Sound Effects)

12. Bibliografia

- The 14 Best Games Developed By Only One Person - GameSpot.* (n.d.). Retrieved December 8, 2020, from <https://www.gamespot.com/gallery/the-14-best-games-developed-by-only-one-person/2900-2172/>
- Practical Unit Testing in Unity3D | by Kuldeep Singh | XRPractices | Medium.* (n.d.). Retrieved December 6, 2020, from <https://medium.com/xrpractices/practical-unit-testing-in-unity3d-f8d5f777c5db>
- Add Multiple Language Support to Your Unity Projects - Unity Forum.* (n.d.). Retrieved November 1, 2020, from https://forum.unity.com/threads/add-multiple-language-support-to-your-unity-projects.206271/?_ga=2.68558097.1012374805.1604261276-1847434671.1594454570
- Working with Physically-Based Shading: a Practical Approach Technologies Blog.* (n.d.). Retrieved October 29, 2020, from <https://blogs.unity3d.com/2015/02/18/working-with-physically-based-shading-a-practical-approach/>
- Paint.NET - Free Software for Digital Photo Editing.* (n.d.). Retrieved October 29, 2020, from <https://www.getpaint.net/>
- Hex Map 1.* (n.d.). Retrieved October 27, 2020, from <https://catlikecoding.com/unity/tutorials/hex-map/part-1/>
- GitHub - jacobdufault/fullserializer: A robust JSON serialization framework that just works with support for all major Unity export platforms.* (n.d.). Retrieved September 27, 2020, from <https://github.com/jacobdufault/fullserializer>
- How to write custom converters for JSON serialization - .NET | Microsoft Docs.* (n.d.). Retrieved September 26, 2020, from <https://docs.microsoft.com/en-us/dotnet/standard/serialization/system-text-json-converters-how-to>
- Hexagonal Grids.* (n.d.). Retrieved September 24, 2020, from <https://www.redblobgames.com/grids/hexagons/>
- Play Hive Like a Champion: Strategy, Tactics and Commentary: Ingersoll, Randy: 9781480060951: Amazon.com: Books.* (n.d.). Retrieved September 22, 2020, from <https://www.amazon.com/dp/148006095X>
- blender.org - Home of the Blender project - Free and Open 3D Creation Software.* (n.d.). Retrieved September 21, 2020, from <https://www.blender.org/>
- Hive en Steam.* (n.d.). Retrieved September 17, 2020, from <https://store.steampowered.com/app/251210/Hive/>

- Hive: La Colmena (juego de mesa) - Aplicaciones en Google Play.* (n.d.). Retrieved September 17, 2020, from <https://play.google.com/store/apps/details?id=com.jb.hive.android&hl=es>
- Hive.* (n.d.). Retrieved September 17, 2020, from <https://www.gen42.com/games/hive>
- Hive - lu2, Experts en jocs de taula.* (n.d.). Retrieved September 17, 2020, from <https://lu2.cat/shop/hive-joc-de-taula/>
- The Quest For A Resolution Independent UI In Unity — TuriyaWare.* (n.d.). Retrieved September 17, 2020, from <https://www.turiyaware.com/blog/the-quest-for-a-resolution-independent-ui-in-unity>
- Draw Freely | Inkscape.* (n.d.). Retrieved September 17, 2020, from <https://inkscape.org/es/>
- Plataforma de desarrollo en tiempo real de Unity | Motor de VR y AR en 3D y 2D.* (n.d.). Retrieved September 17, 2020, from <https://unity.com/es>