



UNIVERSITAT
JAUME I

Upside Downgeon: Development of a videogame based on classic role-playing board games



Javier Albors Caño

Final Degree Work
Bachelor's Degree in
Video Game Design and Development
Universitat Jaume I

June 29, 2022

Supervised by: Diego José Díaz García



To the crew

ACKNOWLEDGMENTS

First of all, I would like to thank my Final Degree Work supervisor, Diego José Díaz García, for his advice and help provided during the development.

Thanks to Oscar and Jordi, who, providing the room and guidance, made possible the dub of the game.

Thank Sergio Barrachina Mir and José Vte. Martí Avilés for their inspiring `LaTeX` template for writing the Final Degree Work report, which I have used as a starting point in writing this report.

Thanks to Sergio, Jose, Boro, Vite, David and my brother for testing the game and providing some crucial advice and for their always welcomed suggestions.

Also, thank you, Alicia, for giving voice to the female characters and your constant care during the development.

ABSTRACT

This document represents the Final Degree Work report of Javier Albors Caño in Video Game Design and Development. Aforesaid work is made up of a PC videogame titled *Upside Downgeon*, a videogame inspired by the classic *dungeon crawler* games.

This videogame seeks to be built as a board game. In this way, the player will find a new way to play a «board game», finding a better immersion thanks to the automation, animation and sound that the electronic basis of the videogame provides. This is the way the project tries to reflect on the relationship, similarities and differences between board games and its electronic descendants.

The project put special emphasis on the game art, sound and appearance but also special care on its game design. Every game asset, including specific scripting, 2D and 3D art, sound and music has been specifically developed.

CONTENTS

Contents	v
1 Introduction	1
1.1 Work Motivation	1
1.2 Objectives	2
1.3 Environment and Initial State	3
2 Planning and resources evaluation	5
2.1 Planning	5
2.2 Resource Evaluation	9
3 System Analysis and Design	11
3.1 Requirement Analysis	12
3.2 System Design	18
3.3 System Architecture	27
3.4 Interface Design	27
4 Game Design	33
4.1 Game concept and world building	33
4.2 Game and mechanics design	36
5 Work Development and Results	41
5.1 Work Development	41
5.2 Results	49
6 Conclusions and Future Work	53
6.1 Conclusions	53
6.2 Future work	54
Bibliography	55
A Diary of Development	57
A.1 Bibliography	60
B Source code and other considerations	61

C Play testing**63**

INTRODUCTION

Contents

1.1	Work Motivation	1
1.2	Objectives	2
1.3	Environment and Initial State	3

The work is, indeed, a complete development of a videogame using several free tools. It was decided that the project would seek to have its own assets specifically developed by the author, so it could be as accurate to the original design as possible. Also, with this, the project wanted to emphasise how much multipurpose the character of this Degree is. The game developed might not be perfect in any discipline of the *game developing*, but it can at least just be good at every of them. This is the report of a multidisciplinary Final Degree Work in an extremely multidisciplinary Degree.

1.1 Work Motivation

Since their appearance, videogames have always been clearly differed from the classic board games in mechanics, design and, obviously, player interaction. Both, board and video games, have found huge evolution and reached new levels of complexity.

Modern *dungeon crawler*[8] board games use new intricate mechanics that usually suppose the need of saving and modifying the game state and character attributes. This makes the players have to take into account a lot of game events, do a lot of math and be extremely careful with what they have on the table in order not to lose or forget what exactly is going on the party and what to do next.

Implementing videogames technologies should help to avoid these issues. This project seeks to develop a complete 3D role-playing board game in which the player will use the

UI and the mouse to interact with the game world (that tries to be as similar to physical board games as possible). The machine will also do the dungeon master's work, this is, managing the game events, mechanics, rules and enemies.

1.2 Objectives

The main goal of the project is creating a virtual environment that simulate a board game party. This virtual environment should exploit every advantage that the tools usually used on videogame development could offer (such as graphic engines, the scripting of calculus automation, sound engines, etc.) in order to ease and improve the user's experience.

That is why the finish of the game, its appearance, sound and player interaction take on an important place on the project's objectives: The players must feel like they are playing a *role-playing board game* and, at the same time, they are playing a videogame with an intuitive tutorial and interface, an entire set of animated characters and a whole original soundtrack.

All of this, to the extent possible. Since the proposal of the project, I had a clear idea that every game asset, including 2D and 3D art, sound and music should be specifically developed for this game. Correctly balancing the time spent in every task was very important.

Furthermore, there were some more other objectives established:

- **User interaction:** Board games are played using a physical support. Players use their hands to pick and move tokens, place tiles, share out tokens, etc. Also they just lean on the board to see and understand what is going on during the game. Here is the first challenge. The interaction should be as simple as leaning yourself over the table and pick and swing a token on a board. While using a computer, the player can not pick a figurine with their hands unless they are playing a VR game.

Also, nowadays, most of the software approach its interface from the interaction with the mouse. It is likely that the player prefers being able to do every single action in the game only by using the mouse. It is similar to use your own hands to pick a cardboard game token.

That is why the users should see every action they can do on the interface, so putting or moving a figurine on a tile is just matter of clicking on the right places of the screen. Also, the mouse should be used on a simple and intuitive way to move the point of view of the player during the game.

- **Game system:** *Role-playing board games* usually have a board compound by several squared tiles. Also, there are usually several different agents or units, that are placed on a single tile and have definite numeric stats (such as health points, speed points, etc.). Some agents would work as enemies and some other would be

worn by the players. These agents move around the board spending movement or speed points that represent the maximum tiles it can move in any direction.

I wanted to include this kind of structure in my game, but I wanted to modify it so the game would be a new and different experience. I also wanted to keep it as simple and intuitive as possible. The game system will be later explained with accuracy.

1.3 Environment and Initial State

The author used to spend countless afternoon playing board games and trading card games since very young. One of the first ideas for developing a game the author had before starting his Design and Development Degree, was making a classic *RPG* board game with similar features that *Upside Downgeon* finally had.

Videogames based on board games do already exist, titles like *Demeo*[2], which has an original game design, or the *Warhammer Blood Bowl*[6], being this one an adaptation from a real board game; but they are very limited in number. The first mentioned one, *Demeo*, is an *RPG* inspired by *Dungeons&Dragons*[5], just as *Upside Downgeon* does. Its game design fits perfectly in the description given in the section 1.2. When the idea of the project came up and the design and development of *Upside Downgeon* started, *Demeo* was, indeed, a VR only game. The idea of moving a similar design and idea to a Desktop PC game seemed interesting and exciting.

The coordinator of the project agreed with the game concept and the idea the author had of specifically developing the whole game. After ending his external practices of the Degree, the author started the development of the game as a full time job.

PLANNING AND RESOURCES EVALUATION

Contents

2.1	Planning	5
2.2	Resource Evaluation	9

This chapter deals with the most technical part of the work. Here the planning and the resources used during the project are presented and explained.

2.1 Planning

The project has been developed according to an incremental method. This means that the work has been developed following an iterative process in which every new stage, reached after every two weeks of work, added new and better features to the game.

During every stage new features had been designed and developed. At the end of the iteration, the current version of the game is tested and shown to the project coordinator. The feedback received would be analyzed and used to design new features for the videogame or fix the current ones, and thus start the development of a new stage. Also, the incremental development requires a deep and constant planning of the goals and subgoals of the project.

Also, every three weeks, a total of two times during the project, a group of real possible users have tested the videogame, providing deep feedback about new features, design and possible errors.

Correctly balancing the time spent in every task during the project was very important. That is why, at the very beginning of the project, a chart was created in order to

collect, on a very simplified way, every great group of tasks and the time that there was expected to spend on them (see Figure 2.1).

This basic planning was modified and developed into a more detailed weekly plan that could be represented using a Gantt chart (see Figure 2.2). There can be seen that, as explained before, the work has been developed as if it was a full time job. It was decided that Mondays would be 4 hours working days, so the work could be combined with 4 hours of classes per week. The rest of the days of the week would be full working days (working a total of 8 hours per day). There was planned to spend a total of 36 working hours per week.

Having that plan developed, the time spent in every single real task during the project would be daily recorded using a project diary (see Figure 2.3). This allows to compare the planning and the real time spent during the project. One of the hardest tasks in which finally has been spent more time than expected is the 3D art. At first, it was expected to model high poly characters in order to, later, retopologize them and paint textures. This was a very time consuming pipeline that ended giving not as good results as expected. Because of that, the high poly model that had been developed until I realized that was finally cast aside and every model was done following a low-poly style. This forced the planning to be changed and reduce the time spend on other tasks. The task that has been reduced at most was the one dedicated to make concept art of the characters (finally, only concept art of the monsters have been done). Consult the whole document in the Appendix A.

Task	Time invested (in hours)
<i>Game design, mechanics and rules.</i>	15
<i>Character and enemies design.</i>	10
<i>Concept art.</i>	10
<i>Model and animation.</i>	90
<i>Sound and music.</i>	10
<i>Setting the assets in the game world.</i>	50
<i>Software implementation.</i>	100
<i>Memory development.</i>	15
Total	300

Figure 2.1: Planning of the project using a simple task chart

Project plan												
Tasks	Start	Finish	Weeks									
			1	2	3	4	5	6	7	8	9	10
			28/03	04/04	11/04	18/04	25/04	02/05	09/05	16/05	23/05	30/05
Gettin started	1	1	8									
First project planning	1	1	4									
ng the Unity 3D project	1	1	2									
source code repository	1	1	2									
Implementation	1	8										
Board	1	2	28	24								
Agents	3	4			16		8					
User interaction	4	5				12	12					
UI	5	6					8	4				
AI	7	8							16	28		
Enviroment	2	4										
Design	2	2		4								
Concept art	2	3		8	4							
Assets	3	4				12						
Characters & monsters	5	7										
Concept art	5	5					8					
Assets	5	6					8	16				
Animation	6	7						16	4			
Sound Design / Production	7	7										
Music	7	7							8	4		
SFX	7	7							8			
Testing	8	8										
Alpha build and test	4	4				4						
Beta build and test	4	4								4		
Testing fixes	12	12										12
Project Memory	9	9										16
Total Project	1	9	36	36	20		36	36	36	36	36	28

Figure 2.2: Planning of the project using a weekly Gantt chart

Tasks	Monday	Tuesday	Wednesday	Thursday	Friday
	04/04	05/04	06/04	07/04	08/04
<i>Spawning units using UI</i>	1 hrs	0	0	0	0
<i>Debugging pathfinding</i>	1 hrs 30 min	0	30 min	0	0
<i>Mana costs and turns system (game rules)</i>	1 hrs 30 min	0	0	0	0
<i>Tooltip to show unit's stats (UI)</i>	0	2 hrs 30 min	0	0	0
<i>Working on UI colors for better contrast</i>	0	30 min	0	0	0
<i>Vertical fog</i>	0	1 hrs 30 min	0	0	0
<i>Moving the camera</i>	0	3 hrs	0	0	0
<i>New board features (can not spawn anywhere)</i>	0	0	2 hrs 30 min	0	0
<i>New agents features (Adding actions and habilities)</i>	0	0	1 hrs 30 min	0	0
<i>Artificial Intelligence for the Adventurers</i>	0	0	3 hrs 30 min	8 hrs	5 hrs
<i>Concept art</i>	0	0	0	0	3 hrs
Total	4 hrs	8 hrs	8 hrs	8 hrs	8 hrs

Figure 2.3: Example of a real recorded week of the project diary document

2.2 Resource Evaluation

In this section are estimated the undertaken costs of the work. The human and equipment costs are quantified so that the work can be assessed and so that, in a real case, the economic viability of the work could be evaluated. Here is a list of the essential and helping software and hardware used during project.

1. **Hardware** valued in a total of **1440€**:

- **HP Pavilion Gaming 15:** Intel Core i7 CPU, 8 GB RAM, Nvidia 1050Ti GPU, valued in 800€.
- **XP-PEN Artist 22 Pro drawing screen:** valued in 500€.
- **BenQ 27" LED IPS screen:** valued in 140€.
- **Universitat Jaume I dubbing equipment:** Not valuable.

2. **Software.** Only free software and online tools have been used:

- **Unity3D Game Engine:** version 2021.3.2f1, free student license.
- **Blender:** version 3.1.2, modeling, sculpting and texturing software, free open source software.
- **Krita:** version 5.0.6, digital painting software, free open source software.
- **GitHub Desktop:** version 3.0.1, free licensed.
- **Overleaf:** online LaTeX edition tool, free licensed.
- **Soundtrap by Spotify:** online sound and music creation tool, free trial subscription (7.99€ per month).

3. **Human resources.** Most part of the work listed here is done by the author of this document. Every role and its work time has been estimated and listed using the development diary document as reference. Every role average salary has been calculated using *Salary finder* from *Indeed.com*[3] data. The salaries found in the Salary finder are often lower than expected, so some numbers have been modified in order not to undervalue those activities. Human resources of the project are valued in total in **3762€**:

- **General project manager:** Author, 52 hours of work. Earning 15€ per hour on average, valued in 780€.
- **Unity 3D developer and junior programmer:** Author, 157 hours of work. Earning 12€ per hour on average, valued in 1884€.
- **2D artist:** Author, 57 hours of work. Earning 12€ per hour on average, valued in 684€.
- **3D artist:** Author, 22 hours of work. Earning 12€ per hour on average, valued in 264€.

- **Sound designer and composer:** Author, 10 hours of work. Earning 12€ per hour on average, valued in 120€.
- **Male voice actor:** Author, 2 hours of work. Earning 10€ per hour on average, valued in 20€.
- **Female voice actress:** Collaborator, 1 hours of work. Earning 10€ per hour on average.

Lacking the value of the dubbing equipment from the UJI, the whole project can be valued in a total of **4313.8€**.

CHAPTER
3

SYSTEM ANALYSIS AND DESIGN

Contents

3.1	Requirement Analysis	12
3.2	System Design	18
3.3	System Architecture	27
3.4	Interface Design	27

This chapter presents the requirements analysis, design and architecture of the proposed work. It also describes the process of design of the GUI (Graphical User Interface, see Figure 3.1).

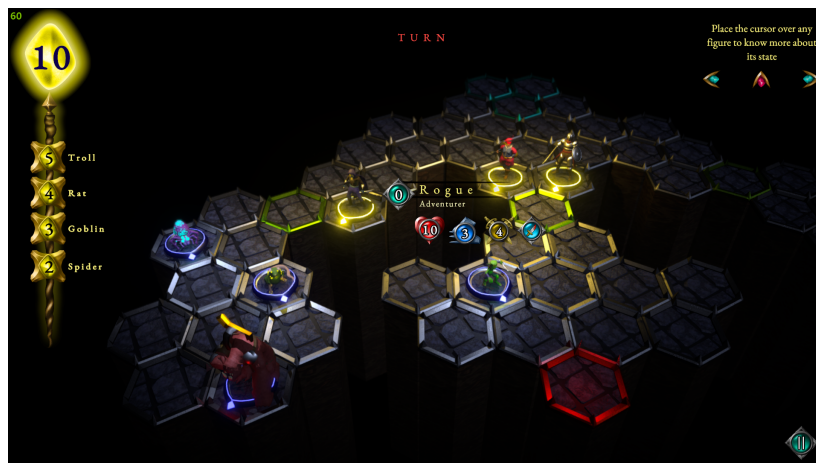


Figure 3.1: Final UI design used in *Upside Downgeon*

3.1 Requirement Analysis

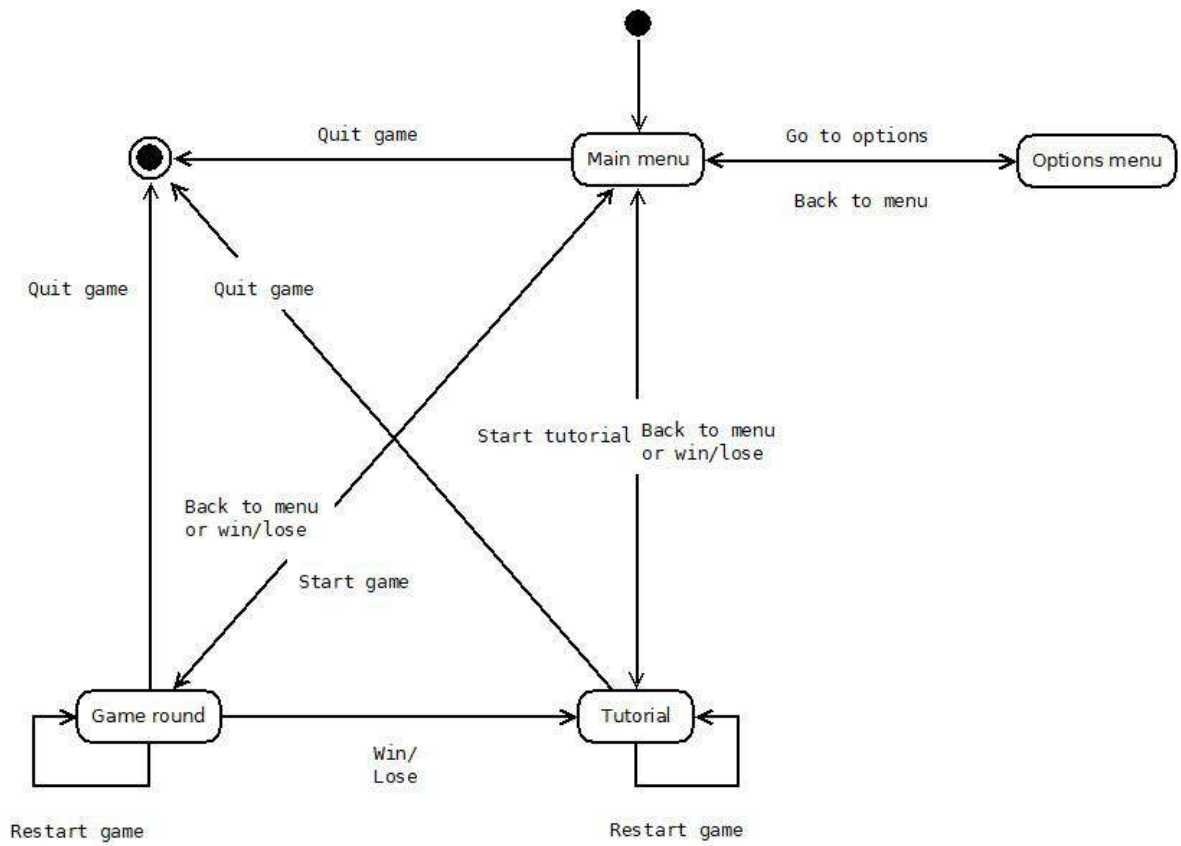


Figure 3.2: Navigation diagram of *Upside Downgeon*

Before enumerating a series of functional requirements, it is convenient to describe and reflect on how the videogame works and leads the user through the gameplay in a preliminary analysis of the requirements of the videogame as software.

Upside Downgeon is a videogame which mechanics are fed by classic «role-playing board games» but implements a *point and click* user interaction. Also, the target of the player is «upside down», as they must not explore a dungeon but defend it. This switches the genre of the game from *dungeon crawler* or *rogue-lite adventure* into a *TBT* (Turn-based tactics) in which the player can spawn and move several units in order to expel the explorers (or *adventurers* as are called i the game).

As the videogame is launched, the main theme of the OST (original soundtrack) can be heard. Also, the user can see an animation that ends showing the main menu screen, that is compound by the game logotype and a total of four buttons (see Figure 3.3). Each one of them says, from top to bottom: «Tutorial», «Start game», «Options» and «Quit». The starting animation of the menu could hide how the system sets the language of the game. As it is English by default, it could be automatically set in Spanish if it is the default language of the computer. *Quit* button, if selected, would close the game screen after an animation. *Tutorial* button would launch a new scene after a fading animation: the tutorial scene. *Start game* button would launch the game scene, similarly to the *Tutorial* button. This button is not available until the user has, at least, selected the *Tutorial* button and entered the tutorial scene.



Figure 3.3: Main menu that can be seen when launching *Upside Downgeon*

Finally, the *Options* button would switch the layout of the main menu into the options layout (see Figure 3.4). The options layout shows a panel that can be used to switch the language of the game (between English and Spanish), a slider to set the sound volume at desire of the user and a «Back» button that would set back the menu layout into the four buttons of the main menu.



Figure 3.4: Options menu of *Upside Downgeon*

As explained at the beginning of this section, *Upside Downgeon* implements a *point and click* user interaction. This means that the user could play just by using the mouse, pointing with the cursor and clicking on the target. During a normal game (not the tutorial), the player could move the POV (point of view) by dragging the mouse cursor to the edges of the game screen. If the cursor is near the upper edge, the POV would move forwards; if it is close to the lower edge, the POV would move backwards; if it is close to one of the side edges, the POV would move towards that same side. These are the classic camera controls that TBT videogames usually implement. The POV can also be moved by using «W», «A», «S» and «D» keys (using the *W* key to move forwards, *A* to move left, *S* to move backwards and *D* to move right). Using these «WASD» camera controls instead of the previously described is optional and the user can decide which one of both control systems to use at any moment. Some users can find using the mouse for every single action during the gameplay as an annoying feature, so using «WASD» keys as camera controls was a feature added so it could lighten the charge of using the mouse and cursor for every action. These controls were also added because they are the most commonly used controls in almost every PC videogame existing.

The player can also click on the buttons of the GUI, which design will be described and explained in the interface design section (see Section 3.4). This could trigger several actions: end the player's turn, show available tiles of the board to *spawn monsters*¹, pause the game or interact with the tips panel.

Ending player's turn will give the control of the POV and the enemy units to the game system, so the player can just observe what the AI is performing. When selecting a monster (friendly unit) to spawn (create an instance on the board), the player can

¹Create instances of units that the player can control on a tile of the game board

select one of the highlighted available tiles where to spawn that unit (see Figure 3.5). Pausing the game would show an in-game menu (see Figure 3.6), showing four different buttons: «Resume», that would close the menu and resume the game; «Restart game», that would reset the game state; «Main menu», that would return to the main menu screen (see Figure 3.3) and «Quit», to close the game screen after an animation.

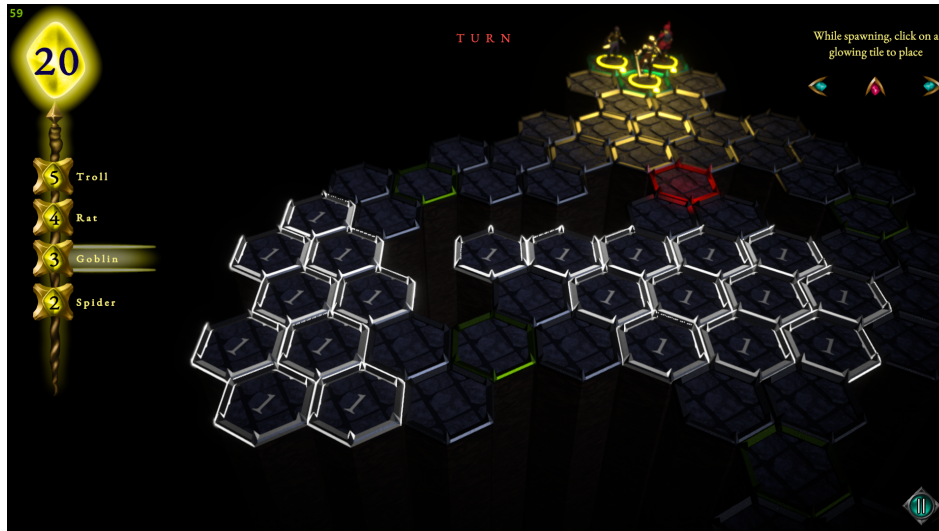


Figure 3.5: Tiles highlighted after clicking on the «Goblin» button



Figure 3.6: In-game menu of *Upside Downgeon*

Finally, the user can use the arrows on the top-right corner of the screen to close or open a panel that shows several game tips or navigate through the list of tips.

During the player's turn, if the player has already spawned monsters during past turns (see Section 4.2), those monsters can be clicked in order to select them. The system will highlight the available targets where the monster can move or attack. If the player clicks on any one of those available targets, the system will calculate and highlight the path, in a different color, that the monster will follow (jumping from one to another tile of the board) to reach it (see Figure 3.7).

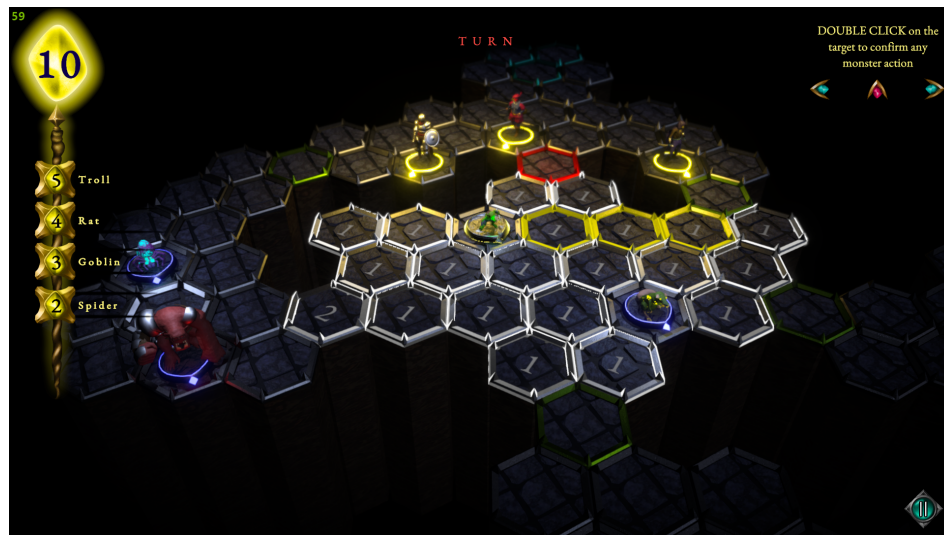


Figure 3.7: Highlighted available tiles where the goblin can move

At the point of having a unit selected and a target for that unit, the player can decide whether to move the unit towards that target by clicking again on it, to select a different target by clicking on it, to deselect the unit by clicking on it or to select one different unit.

All the described actions are required so that the player can interact with the videogame system, and so, play it. Now let's proceed to enumerate the functional and non-functional requirements.

3.1.1 Functional Requirements

- **R1.** The player can play the tutorial.
- **R2.** The player can play a new game.
- **R3.** The player can enter the options menu.
- **R4.** The player can decide the language of the game.
- **R5.** The player can decide the general sound volume of the game.
- **R6.** The player can go back to the main menu.

- **R7.** The player can quit and close the game.
- **R8.** The player can move its point of view.
- **R9.** The player can select a unit to place on the game board.
- **R10.** The player can select a tile where to place a unit.
- **R11.** The player can end their turn.
- **R12.** The player can select a friendly unit on the game board.
- **R13.** The player can select a target for the selected unit.
- **R14.** The player can move the selected unit.
- **R15.** The player can deselect the unit.
- **R16.** The player can open an in-game menu.
- **R17.** The player can reset the game state.
- **R18.** The player can close the in-game menu and resume the game.
- **R19.** The player can hide or show the tips panel.
- **R20.** The player can set the next or previous tip on the tips panel.
- **R21.** The system can move the player's point of view.
- **R22.** The system can select enemy units.
- **R23.** The system can select a target for the selected unit.
- **R24.** The system can move the selected unit.
- **R25.** The system can end its turn.

3.1.2 Non-functional Requirements

- **R26.** The main mechanics can be learnt during the tutorial.
- **R27.** The game will be playable using just the mouse on a computer.
- **R28.** There must be different unit types of both sides (friendly and enemy sides).
- **R29.** Any unit's side must be clear at a glance.
- **R30.** There must be different tile types in the board.
- **R31.** The board must, at least, be compound of four different rooms.
- **R32.** The music will change depending on which side (the player or the system) is performing its turn.

3.2 System Design

This section presents the logical design carried out by the videogame system. To represent the logical design on a visual way, it was created a cases of use diagram (see Figure 3.8). Every case of use represented in the diagram is explained in the charts below (see Charts 3.1 to 3.20).

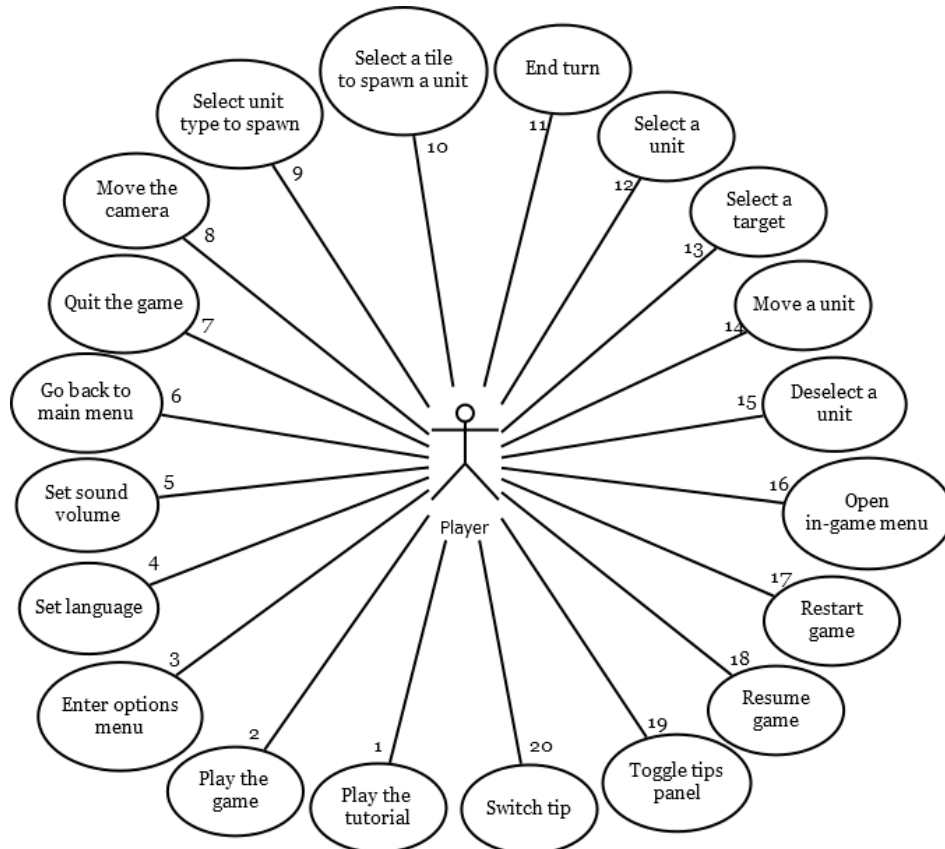


Figure 3.8: Cases of use diagram (Made using Dia)

Requirement:	R1
Actor:	Player
Description:	The player starts the tutorial by clicking on the <i>Tutorial</i> button on the main menu.
Preconditions:	1. The player is on the main menu.
Normal sequence:	1. The player clicks on the the <i>Tutorial</i> button on the main menu. 2. The tutorial scene starts after an animation.
Alt. sequence:	None.

Table 3.1: Case of use «CU01. Play the tutorial»

Requirement:	R2
Actor:	Player
Description:	The player starts a normal game by clicking on the <i>New game</i> button on the main menu.
Preconditions:	1. The player is on the main menu.
Normal sequence:	1. The player clicks on the the <i>New game</i> button on the main menu. 2. The game scene starts after an animation.
Alt. sequence:	None.

Table 3.2: Case of use «CU02. Play the game»

Requirement:	R3
Actor:	Player
Description:	The player enters the options menu by clicking on the <i>Options</i> button on the main menu.
Preconditions:	1. The player is on the main menu.
Normal sequence:	1. The player clicks on the the <i>Options</i> button on the main menu. 2. The main menu buttons are hidden and the option panels are shown after an animation.
Alt. sequence:	None.

Table 3.3: Case of use «CU03. Enter options menu»

Requirement:	R4
Actor:	Player
Description:	The player sets the language of the game using the <i>Language</i> panel of the options menu.
Preconditions:	1. CU03 - Enter options menu.
Normal sequence:	1. The player clicks on the <i>Language</i> panel to switch it from one to the other. (there is only English and Spanish available).
Alt. sequence:	None.

Table 3.4: Case of use «CU04. Set the language»

Requirement:	R5
Actor:	Player
Description:	The player sets the general volume of the game using the <i>Volume</i> slider of the options menu.
Preconditions:	1. CU03 - Enter options menu.
Normal sequence:	1. The player clicks and slides the handler to the desired level of volume.
Alt. sequence:	None.

Table 3.5: Case of use «CU05. Set the sound volume»

Requirement:	R6
Actor:	Player
Description:	The player enters the main menu by clicking the associated button.
Preconditions:	1. CU03 (the player uses the options menu <i>Back</i> button) or CU15 (the player uses the in-game menu <i>Main menu</i> button).
Normal sequence:	1. The player clicks on the <i>Back</i> button of the options menu.
Alt. sequence:	1. The player clicks on the <i>Main menu</i> button of the in-game menu.

Table 3.6: Case of use «CU06. Go back to the main menu»

Requirement:	R7
Actor:	Player
Description:	The player closes the game window by clicking on a <i>Quit</i> button.
Preconditions:	1. CU03 (the player uses the main menu) or CU15 (the player uses the in-game menu).
Normal sequence:	1. The player clicks on the <i>Quit</i> button of the options menu.
Alt. sequence:	1. The player clicks on the <i>Quit</i> button of the in-game menu.

Table 3.7: Case of use «CU07. Quit the game»

Requirement:	R08
Actor:	Player
Description:	The player moves the POV towards a direction.
Preconditions:	1. CU01 or CU02.
Normal sequence:	1. The player drags the mouse cursor to a corner of the screen. 2. The system moves the game camera towards the associated game world cardinal.
Alt. sequence:	1. The player presses «W», «A», «S» or «D» keys. 2. The system moves the game camera towards the associated game world cardinal.

Table 3.8: Case of use «CU08. Move the camera»

Requirement:	R09
Actor:	Player
Description:	The player clicks on a spawn button (with the name of a monster) to trigger spawning sequence.
Preconditions:	1. CU01 or CU02.
Normal sequence:	1. The player chooses a monster type to spawn and clicks on its button. 2. The system highlights the available tiles.
Alt. sequence:	None.

Table 3.9: Case of use «CU09. Select a unit type to spawn»

Requirement:	R10
Actor:	Player
Description:	The player clicks on a highlighted tile to spawn the previously selected monster type.
Preconditions:	1. CU09 - Select a unit type to spawn.
Normal sequence:	1. The player clicks on the desired highlighted tile. 2. The system creates an instance of the unit on the selected tile.
Alt. sequence:	None.

Table 3.10: Case of use «CU10. Select a tile to spawn a unit on it»

Requirement:	R11
Actor:	Player
Description:	The player clicks on the <i>Turn</i> button to end their turn and start the enemies' turn
Preconditions:	<ol style="list-style-type: none"> 1. CU01 or CU02. 2. The game is waiting for the player to perform their turn (the game is not ended and the system is not performing the enemies' turn).
Normal sequence:	<ol style="list-style-type: none"> 1. The player clicks on the <i>Turn</i> button. 2. The system performs the enemies' turn (AI).
Alt. sequence:	None.

Table 3.11: Case of use «CU11. End turn»

Requirement:	R12
Actor:	Player
Description:	The player selects a friendly unit (<i>monster</i>) by clicking on it
Preconditions:	<ol style="list-style-type: none"> 1. The player is on a game scene.
Normal sequence:	<ol style="list-style-type: none"> 1. The player clicks on the desired unit. 2. A sound and an animation plays. The system highlights the available targets.
Alt. sequence:	None.

Table 3.12: Case of use «CU12. Select a unit»

Requirement:	R13
Actor:	Player
Description:	The player selects a highlighted enemy unit (<i>adventurer</i>) or tile by clicking on it.
Preconditions:	<ol style="list-style-type: none"> 1. CU12 - Select a unit.
Normal sequence:	<ol style="list-style-type: none"> 1. The player clicks on the desired highlighted foe or tile. 2. The system highlights the tiles of the path that the friendly unit selected during CU12 would follow towards the selected target on a different color.
Alt. sequence:	None.

Table 3.13: Case of use «CU13. Select a target»

Requirement:	R14
Actor:	Player
Description:	The player clicks on a previously selected tile by clicking on it. This will trigger the move of the selected unit. If the selected target is an enemy unit, the selected friendly unit will move towards it and attack.
Preconditions:	1. CU13 - Select a target.
Normal sequence:	1. The player clicks on tile on which clicked during CU13. 2. The system moves the friendly unit selected during CU12 towards the target.
Alt. sequence:	1. The player clicks on the foe on which clicked during CU13 (target tile or foe). 2. The system moves the friendly unit selected during CU12 towards the a neighbour tile and perform an attack (a sound and an animation would be played).

Table 3.14: Case of use «CU14. Move a unit»

Requirement:	R15
Actor:	Player
Description:	The player clicks on a previously selected friendly unit by clicking on it. This will deselect that unit and reset the highlighted tiles.
Preconditions:	1. CU12 - Select a unit.
Normal sequence:	1. The player clicks on the friendly unit selected during CU12. 2. The system resets the tiles so the highlighted tiles return to its normal appearance.
Alt. sequence:	None.

Table 3.15: Case of use «CU15. Deselect a unit»

Requirement:	R16
Actor:	Player
Description:	The player clicks on the <i>Pause</i> button of the GUI. This will show the in-game menu and its several options
Preconditions:	1. CU01 or CU02.
Normal sequence:	1. The player clicks on the <i>Pause</i> button. 2. The system shows the in-game menu and its several options.
Alt. sequence:	None.

Table 3.16: Case of use «CU16. Open the in-game menu»

Requirement:	R17
Actor:	Player
Description:	The player clicks on the <i>Restart</i> button of the in-game menu. This will reload the game scene so the game state will be reset
Preconditions:	1. CU16 - Open the in-game menu.
Normal sequence:	1. The player clicks on the <i>Restart</i> button. 2. The system reloads the game scene so the game state would be reset.
Alt. sequence:	None.

Table 3.17: Case of use «CU17. Restart the game»

Requirement:	R18
Actor:	Player
Description:	The player clicks on the <i>Resume</i> button of the in-game menu. This will hide the in-game menu and resume the game
Preconditions:	1. CU16 - Open the in-game menu.
Normal sequence:	1. The player clicks on the <i>Resume</i> button. 2. The system hides the in-game menu.
Alt. sequence:	None.

Table 3.18: Case of use «CU18. Resume the game»

Requirement:	R19
Actor:	Player
Description:	The player clicks on a button of the tips panel on the GUI. This will toggle (show if hidden or hide if visible) the tips panel
Preconditions:	1. CU01 or CU02.
Normal sequence:	1. The player clicks on the <i>hide/show</i> button of the tips panel. 2. If the tips panel is hidden, the system animates the tips panel to hide it.
Alt. sequence:	1. The player clicks on the <i>hide/show</i> button of the tips panel. 2. If the tips panel is hidden, the system animates the tips panel to show it.

Table 3.19: Case of use «CU19. Toggle tips panel»

Requirement:	R20
Actor:	Player
Description:	The player clicks on the <i>Next</i> or the <i>Previous</i> button of the tips panel on the GUI. This will change the tip shown on the tips panel.
Preconditions:	1. CU01 or CU02. 2. The tips panel is visible.
Normal sequence:	1. The player clicks on the <i>Next</i> button of the tips panel. 2. The system animates the tip panel and correctly switches the tip shown on it.
Alt. sequence:	1. The player clicks on the <i>Previous</i> button of the tips panel. 2. The system animates the tip panel and correctly switches the tip shown on it.

Table 3.20: Case of use «CU20. Switch tip»

3.3 System Architecture

The minimum requirements to play the build of the project are:

- Operative System: Windows 7, Windows 10 or Windows 11.
- CPU: x86 or x64 architecture with SSE2 instruction set support.
- GPU: nvidia 940MX.
- Memory on disk: 200 Mb.
- Peripherals: mouse or touch panel.

For a better comfort, it is advisable to use mouse and keyboard.

3.4 Interface Design

The Since there has been made a total of two playtest, having collaborated a total of 10 players and the answers about the UI distribution have always been good, the original GUI design of *Upside Downgeon* has almost kept the same structure from the beginning of the project (see Figure 3.9).



Figure 3.9: First complete GUI framework of *Upside Downgeon*

Bearing in mind the design of the game (see Section 4.2), there were some essential elements that must be taken into account at the time of designing the GUI framework. Those elements were: the «Mana panel», a tip box to show game tips for the player as banners, a button to end the player's turn and a button to show an in-game menu. Also, an essential feature of the game is a tool-tip box that shows info whenever the player drags the cursor over a unit (enemy or not). This last element has encountered the biggest change since its first design.

The *Mana panel* (top left corner) shows info about the actions related with the game rules that the player can take. This panel shows the *Mana Points* of the player to spend when spawning friendly units and a group of buttons. These are the *Spawn buttons*. Each of those buttons contain a name (the name of the unit to spawn) and a number (its cost in *Mana Points*). It was not necessary for the player to know a lot about the units at the beginning of the game, as a lot of players could find it interesting to explore their possibilities on their own.

The *Mana panel* takes up a lot of screen space. This could make some users uncomfortable. It was decided that, as *Spawning buttons* are not used during some stages of the game (during tutorials or enemy turn) it should be hidden with its proper animation. Even so, the current *Mana Points* of the player can still be important information during all the game, that is why it was decided to keep it during enemies' turn (see Figure 3.10).



Figure 3.10: Final GUI design during the enemies' turn

Ending the turn is also an action related with the game rules, that is why, on the first design of the GUI, the *End turn* button was under the *Mana panel*. Mainly for artistic reasons, that button ended leaving its original position. On a middle stage, it was placed at the lower center of the screen (see Figure 3.11). Some users found this annoying, as they clicked on it by accident when spawning monsters on the board. Top center of the screen seemed the best option, as the top part of the screen contains less game world information due to the point of view of the player, so that is its final position.

As it can be seen on the original GUI framework (see Figure 3.9), the tool-tip box shows info about units on a textual way. Later, it was decided that the stats of the units must be shown as icons with their numeric value (see Figure 3.12). Also, every skill of the units should be represented with an icon. It was important that the icons had different shapes and colors so it could help users identify every stat or skill at a glance.

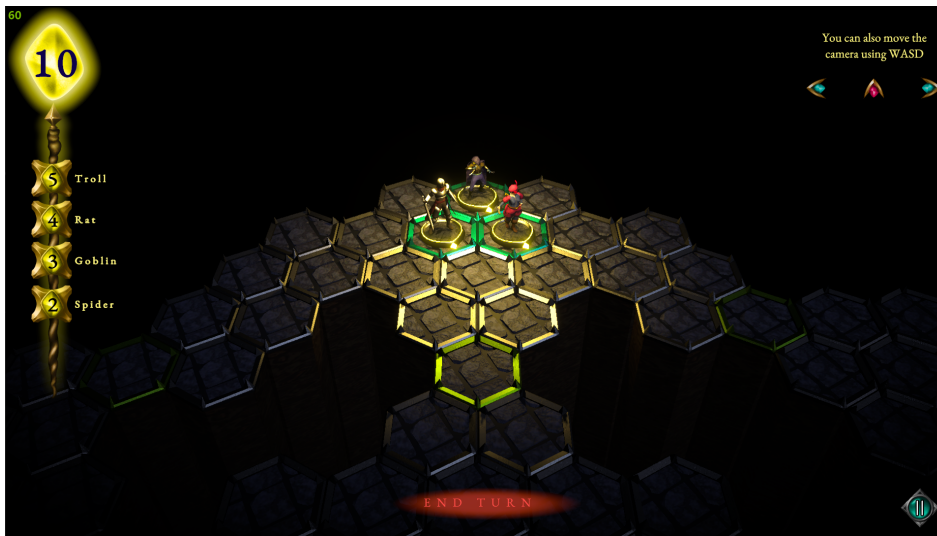


Figure 3.11: GUI of a middle stage of *Upside Downgeon* development



Figure 3.12: Detail of the tool-tip box used in *Upside Downgeon*

Finally, every tutorial stage has its original and specific designed minimal GUI. Almost every stage hides the usual GUI of the game and show a text on the most convenient part of the screen for the item or mechanic it is showing (see Figure 3.13). Some stages can also show different parts of the usual GUI (see Figure 3.14) or some icons (see Figure 3.15). At any moment of the tutorial, the player can skip it and continue with the game normally by clicking on the «Skip» button. This button is shown in the usual position of the *End turn* button. The player will navigate through the tutorial by clicking on any part of the screen or following the instructions that the stage could be giving.



Figure 3.13: Layout of the tutorial showing highlighted tiles

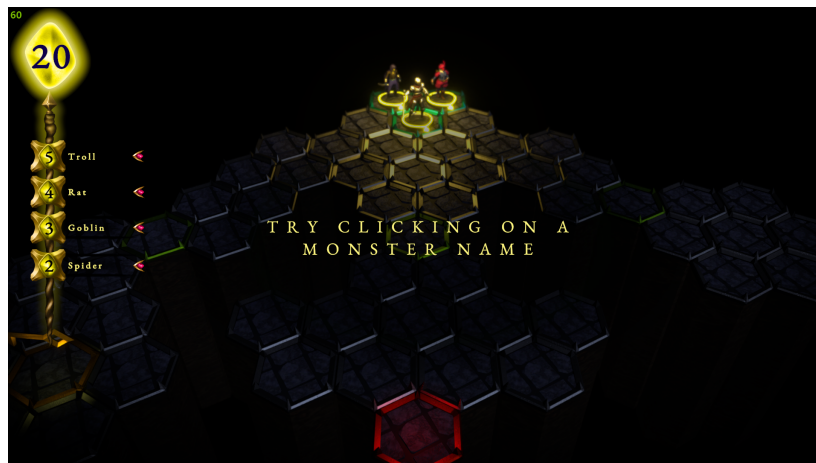


Figure 3.14: Layout of the tutorial showing the *Spawning buttons*



Figure 3.15: Layout of the tutorial showing an icon

CHAPTER 

GAME DESIGN

Contents

4.1	Game concept and world building	33
4.2	Game and mechanics design	36

In this chapter is presented the game design. During the following sections, the game concept, the character and level design, the agents and the mechanics that implement the videogame are detailed and explained.

4.1 Game concept and world building

Upside Downgeon is a role playing turn-based board game. The game will take place in 2D boards composed of hexagon-shaped tiles. Its aesthetic and design will be based on the classic dungeon crawler board games. These games are usually set in a dark fantasy medieval world inspired by Tolkien's work of literature and the theme of *Dungeons & Dragons*.

The game world in which *Upside Downgeon* is set is a classic fantasy world, full of fantastic creatures, curses, evil warlocks and intrepid adventurers. The only issue is that the objective in this adventure is not «crawling the dungeon» but trying to keep it shut tight. The player will not embody any warrior, archer, wizard nor rogue, but its role will be the one of the evil sorcerer that leads the wicked horde of creatures that scrape by in the darkness of the dungeon.

The game will take place in an ominous underground dungeon in which latest chamber rests the treasure of the malignant unmentionable *Warlock*, known as *Gorthrone*. There, the evil sorcerer keeps countless magical artifacts and materials, cursed weapons

and his ancient and magnificent mana font. Before reaching the bottom of the dungeon, any adventurer who dares trespassing its doors will have to survive countless battles against the army of darkness that dwell in its dreadful chambers.

4.1.1 Characters

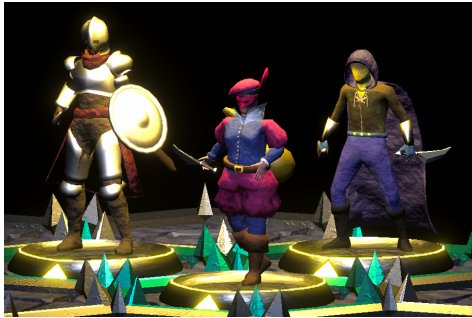


Figure 4.1: Detail of the Adventurers in game

heavy armor made of iron, leather and brass, as well as her delicate manners, make clear her noble ancestry. The coat of arms on her breastplate has been totally uprooted. She had lost her right to the throne after escaping from the lands of her father. What could make her take that decision? Is she just following her own way of life or is there a darker reason for her to decide she would follow her own path?

The bard is like the most enthusiastic of the bards. She thinks that she does not need his part of the loot because her experience during their adventures will make her write the best songs of the *Human Realms*. She is old, slow, heavy and weak, but even an old pirate and a renegade noble can appreciate how her good music vibes during the battle. She wants to know everything about her mates in search of inspiration, but she is so quiet about her own story... Just don't think too much about it, she is just a good bard.

They might seem interesting and friendly characters but don't forget that they are the enemy. The three of them want to steal the warlock's treasure and destroy his *Mana* font. The player will play as the *Warlock*, leading the monsters in order to make the adventurers desire to have never tried to enter the dungeon.

Goblins are little wicked greenish humanoids. They have huge ears compared to their emaciated bodies. They are weak, but light, silent and nimble so they are extremely harmful when using the element of surprise.

Trolls are extremely heavy and strong creatures. They also stand on two feet but you can hardly say they are man-shaped. Their massive arms, pointy fangs and hollow

The rogue is an old bounty hunter that once upon a time led a pirate crew. He is an extremely nimble man for his age that joined the other two adventurers presumably just to claim his part of the loot. But if things get messed up during the raid into the dungeon, he could surprise his mates with an extreme sense of honor and give his best and risk his own life in order to complete the mission safe and sound. Is it that he is truly willing to protect his new crew or he just thinks that the success depends on the entirety of the group?

The warrior is a true adventurer. She is a young woman who has been raised as the first-born son of a feudal lord. Her

eyes would make anybody who ventures into the dungeon in search of wealth just run as far as their feet allow.



Figure 4.2: Detail of a Goblin and a Troll in game

The *spiders* in the dungeon are just inexplicably huge. Their sting could paralyze the robustest of the warriors. Similarly, the enormous and sick *rats*, with their filthy fur and nauseating pustules could easily make sick and poison any of the adventurers.



Figure 4.3: Detail of a Goblin and a Troll in game

4.2 Game and mechanics design

4.2.1 Level design

Dungeon Keepers is a role playing turn-based board video game. There is a single playable adventure and a tutorial (with a simplified board and weaker enemies). The player will fight against the *Adventurers* that are trying to raid the dungeon across a series of 2D boards composed of hexagon-shaped tiles (see Figure 4.4).



Figure 4.4: Screenshot of the board of the game with some agents on

The boards will represent the floor of the dungeon. The shape of the level must be irregular, finding different rooms and corridors separated by walls or void. The board of each level (tutorial and normal game) will be specifically designed. The player will see and know the shape of the board, where every agent is placed and where is the way to the next room at every moment of the party.

The space in which the game takes place can be hierarchically divided: **boards** are composed of **rooms**; rooms are a series of **tiles** isolated from other rooms by void and connected by **doors**.

At the beginning of every party, the dungeon is clear of monsters and the adventurers are placed on the *entrance tiles*. There is also an ending door (to the next level or the end of the game) and a key. Each one of these elements are on a single different tile somewhere in the board of the level of the dungeon.

4.2.2 Agents of the game

There are three big types of agents: the adventurers, the monsters and the nests. There is also an item: the key, and a special tile: the door. There are three different adventurers: the rogue, the warrior and the bard. There are four different types of monsters: the

troll, the goblin, the rat and the spider. There are two different types of nest: spider nest and rat nest. We will see their attributes and actions in detail.

Attributes

There will be three kinds of adventurers and four kinds of monster that represent every of the characters described in the Characters section (see Section 4.1.1). Every unit will have five attributes with different values depending on the character: position, orientation, speed, health, melee damage and action points.

- The **position** will be the tile of the board where the adventurer is on.
- The agents will be always looking (**orientation**) to the center of a side of the tile in which they are. The orientation is defined by the opposite orientation (180°) of the one looking to the center of the side which the agent came from to the current tile during its last move (see Main mechanics section).
- The **speed** is an integer from one to four that means the number of times the adventurer can jump from a tile to another during its turn.
- Every agent will have a defined number of **health** points (HP). Whenever an agent has lost enough HP by receiving damage (see Main mechanics section), it will disappear and free the tile which was placed on.
- The **melee damage** is low value integer that, normally, indicates how much HP will lose the target of an attack of this agent.
- Every agent have a total of two **action points**. These are spent in moving, attacking or moving and attacking (using one single action point).

Actions and skills

Every unit can move, perform an attack and pick the key. They also might have a passive skill.

As explained, an agent can move as far as its speed attribute allows. If an enemy is on a tile in the range of movement of that agent or on a neighbour tile of a tile in range, this agent could be selected as objective of an attack. This will move the unit and attack using a single action point. On the other hand, the agent could be placed on a neighbour tile to that enemy in range and, after that, select that enemy as objective of an attack. This would be considered as two different actions, and so, spend two action points.

The effects of an attack from one to another agent could vary depending on the passive skill of the unit attacking. Units with high HP usually do not have any passive skill (such as the *Warrior* and *Trolls*). This units perform normal attacks, that just decreases target's HP in the same quantity of the melee damage value of the attacker.



Figure 4.5: Icons for skills and stats used in the game

Some other weaker units have passive skills in order to add diversity of effects to the gameplay.

These skills are:

- **Paralyse:** Spiders have this skill that, using a chance of 50/50, similar to a coin flip, can apply an effect that makes the target of the attack unable to move to a different tile during the next turn (as if its speed was null). Units affected by paralyse can still attack neighbour enemies.
- **Poison:** Rats have this skill that, using a chance of 50/50, similar to a coin flip, can apply an effect that makes the target of the attack lose 1 HP during the next two turn shifts. If the affected unit is at 1 HP, this effect will not be applied (no unit can be killed by poisoning).

- **Stab:** *Rogue*, *Bard* and *Goblin* have this skill. A unit with this skill will apply twice the damage to its target if the attack is performed «from the back». A unit has three front sides and three back sides. Every unit has a front side indicated by a glowing triangle on its base. It is considered that an attack has been performed from the back if the attacker is looking at one of its target's back sides (the opposite side to the front side or its adjacent sides).

Special agents

In every board, there is a key and a door to the next level. The key is an agent placed on a tile on the board. This agent can not perform any action. To end the level (and so the player would lose), one of the adventurers must «open the end door». Opening the door means placing on the *End door tile* bearing the key. In this way, there is an action more to add to possible actions that a unit can do: picking-up an item (key). Monsters can also pick-up the key, but they can only carry it (it would not make sense if the monsters would open that door for the adventurers). Once the key has been picked-up by any unit, it will leave the board for the rest of the game. Any time that a unit with the key is killed, the key possession would be transferred to its killer.

4.2.3 Game mechanics

The game will start at the entrance of the dungeon, starting with the turn of the player. The adventurers will start on the entrance tiles (always three different adventurers, one per type). The player will start with a total of 20 *Mana points* and the dungeon will be clear of monsters.

The player's turn

The player can spawn as many monster as their Mana points quantity allow. Every spawning action creates a single friendly unit on the board. Every spawning action, called *spell*, has a “mana” cost. The player will start the party with 20 *Mana points*. 5 *Mana points* will be restored at the beginning of the next player's turn. After the first player's turn, the player can stack up to 10 *Mana points*. The possible effects of the *spells* are the following:

- **Summon spider:** Creates a spider on a desired tile. The *mana* cost is the lowest (2).
- **Summon goblin:** Creates a goblin on a desired tile. The *mana* cost is low (3).
- **Summon rat:** Creates a rat on a desired tile. The *mana* cost is medium (4).
- **Summon rat:** Creates a rat on a desired tile. The *mana* cost is high (5).

These summoning spells can be cast at any time during the player's turn. Also, they only can be cast on a tile in an **empty room**. That means that there is no adventurer in that room. Rooms are a series of tiles isolated from the others by *void* (see Level Characteristics). Also, as Door tiles connect several rooms, the player can not spawn monsters on the nor their neighbours.

During the player's turn, the player can also move the monsters once. This means that the monster can move as far as its speed attribute indicates. If the speed attribute is 1, the monster can just move to an adjacent tile; if it is 10, it could jump up to ten tiles far. The player must move the monsters to attack the adventurers in order to kill every of them. The player will win the game at any moment that the board is clear of adventurers.

Whenever the player is over with their moves and spells, can end the turn. Then, between turn events (poison) might occur and the adventurers' turn will start.

The adventurers' turn

Adventurers are led by the computer. Their turn is similar to the player's turn, but they can only move and attack. The AI will just take into account the elements in the room the adventurer is in and do not know where the *Key* nor the *End door* is. The objective of the AI is to explore the dungeon in search of the key and the door to open it as soon as possible.

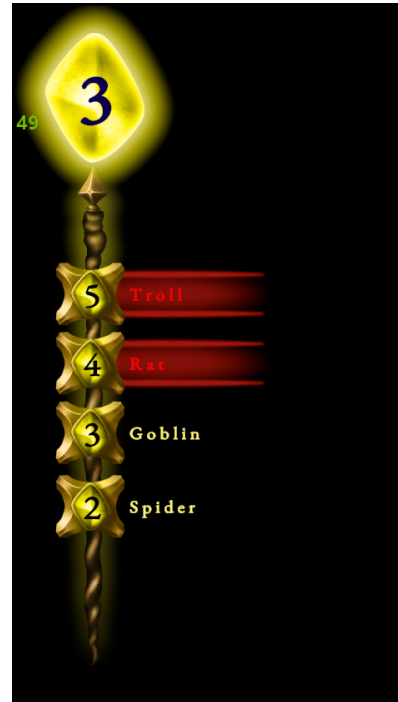


Figure 4.6: Detail of the GUI used to spawn monsters in the game

WORK DEVELOPMENT AND RESULTS

Contents

5.1	Work Development	41
5.2	Results	49

This chapter presents the developed work and the obtained results. Also, the deviations from the initial design are detailed and justified.

5.1 Work Development

This section explains the development of the project in chronological order. The development has been done by two-week long sprints. During sprints, group of tasks from the original planning were developed. For information about the specific tasks and the time spent in every of them, the diary of development can be consulted (see Appendix A).

5.1.1 First sprint

In this section is described the work done during the first two weeks. During this first sprint, the whole basic game system was developed. It can be said that during the first sprint, most part of the technical development was done.

First, the Unity3D project was created. At first, the game was titled *Dungeon Keepers*, as the objective of the player during the adventure is not “crawling the dungeon” but trying to keep it shut tight. It is, indeed, a very similar title to the one of an existing videogame by *Electronic Arts*[1]. This is an old and famous title, so, later, the author would decide to change the title of the game to *Upside Downgeon*, which is a fun word game that seems not yet exist as a videogame title.

During this sprint, it was also created the first detailed project planning (see Figure 2.2), the Unity3D project that was used to develop the game and the code repository on GitHub. There were also created the first 3D models of the game. They were a simple hexagonal tile, a hexagonal figure used as units (see Figure 3.9) and a simple key. Each different type of tile and unit was differed by its color. Every of them were used as placeholders, so none of them ended being used in the final version of the game. The key model would be later be improved and is currently being used.

The first interaction events were created, such as detecting clicks on tiles. Shader graphs were created in order to make the tiles glow and give visual feedback to the user (see Figure 5.1). Those shader graphs are the ones that are still being used to highlight the tiles. After that, it was decided to upgrade the project into the Unity DH render pipeline, but the shaders stopped working and spending time in learning how to remake them with the new render pipeline, i decided to go back to Universal Render Pipeline. That is the pipeline that is still being used on the final version (with some features added).

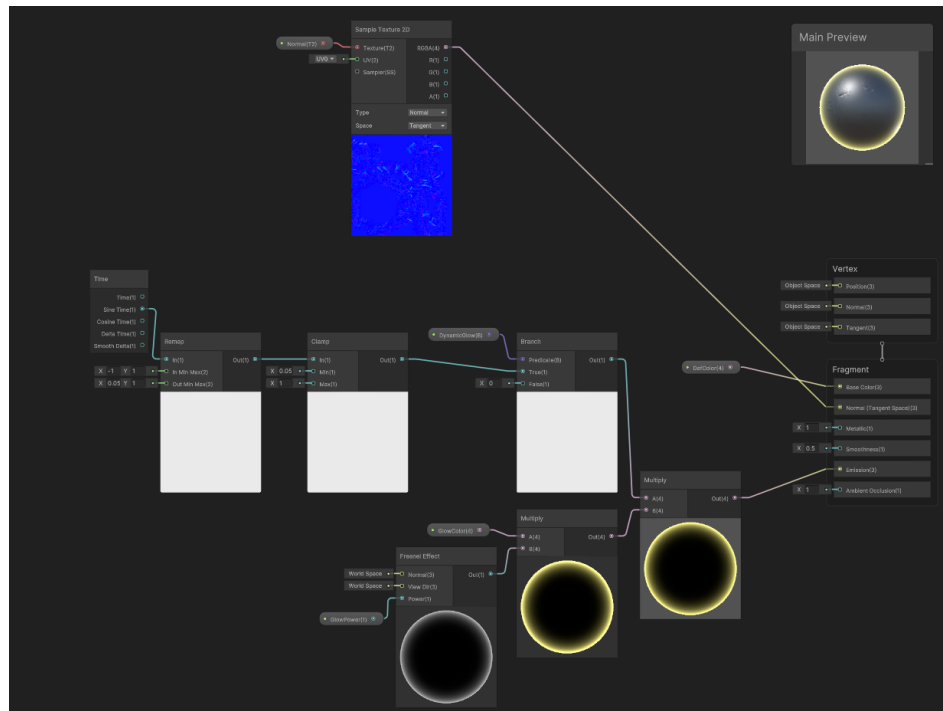


Figure 5.1: Highlight Shader Graph used on *Upside Downgeon*

As explained during the last chapter, the game uses an hexagonal grid as board. A simple board was created, and, also, a system that automatically transformed the 3D world position of every tile into its hexagonal grid coordinates. There were created agents that could be moved by the board tiles by the a user. The user interaction system of moving units is the same moving system that is currently being used in the game. Agents use a path finding algorithm based on a Breath First Search.

Also the first game rules were implemented. As a turn-based tactics game, units of both sides were created. Using a simple GUI, the user could switch from the turn of one of the sides to the other's turn. During each turn, the proper agents could be moved. Adventurers could pick the key and the game could be won by stepping on the *End door* using an Adventurer prototype during Adventurer's turn. If, a Monster killed every of the foes, the game could be won during *Monster's* turn. To achieve this, a picking system that replaced the key by a normal tile was done. Also, the system turned the *End door* tile into walkable or not depending on the unit selected. An unit could step on the *End door* only if it is an Adventurer carrying the key.

During the second week of the development, it was created the first GUI layout and a system to allow the player spawn several Monsters during Monster's turn on the available tiles (see Section 4.2.3). Also the mana system was implemented, the tool-tip box system was created (see Section 3.4) and a system to allow the player move the POV using the mouse cursor position on the screen was done. The new Unity Input System[7] was used to allow the player move the POV also using keyboard (see Figure 5.2).

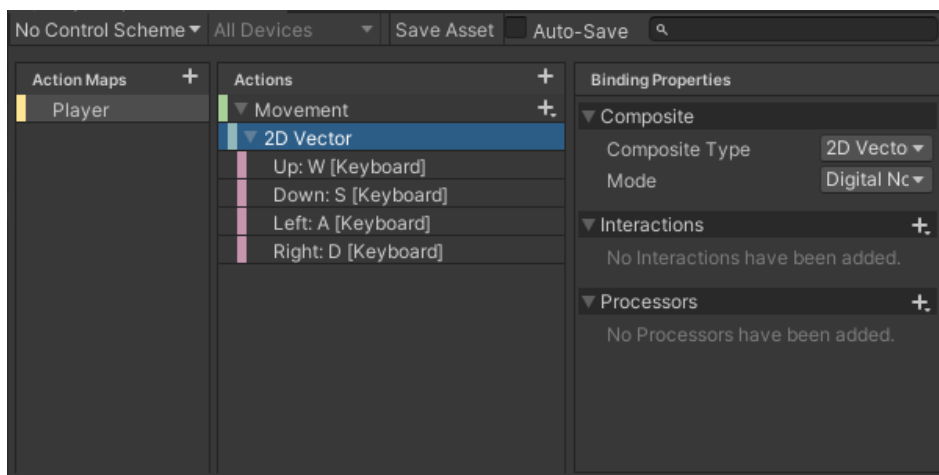


Figure 5.2: Player actions on the new Unity Input System

Artificial intelligence of the Adventurers

It also started the work on the adventurers AI. Until that moment, the player could move both, Monster and Adventurer units, during the proper turns. Now, a *Unit Managing* agent would manage the Adventurers during their turn.

Every adventurer has a *brain* that preforms an action whenever the Unit Manager asks for it. For the AI, performing an action means executing a Binary Decision Tree (see Figure 5.3) that ended spending an action point. It was decided to use this kind of behavior representation because it was not necessary to use a more complicated method, as the game is a very simple TBT game. Both, Unit Manager and the Adventurer's brain, implement *coroutines* in order to make the player see what is going on during the enemy turn.

Using this AI, the enemies must explore the dungeon in search of the key and the end tile while trying to survive to the actions of the player. To achieve this on a quick way, every Adventurer «remembers» (saves data about) the rooms that has already visited and its content. This was implemented in order to create the illusion that they remember where the *End door* tile is if they have already visited the room where it is, but have not the key in their possession yet.

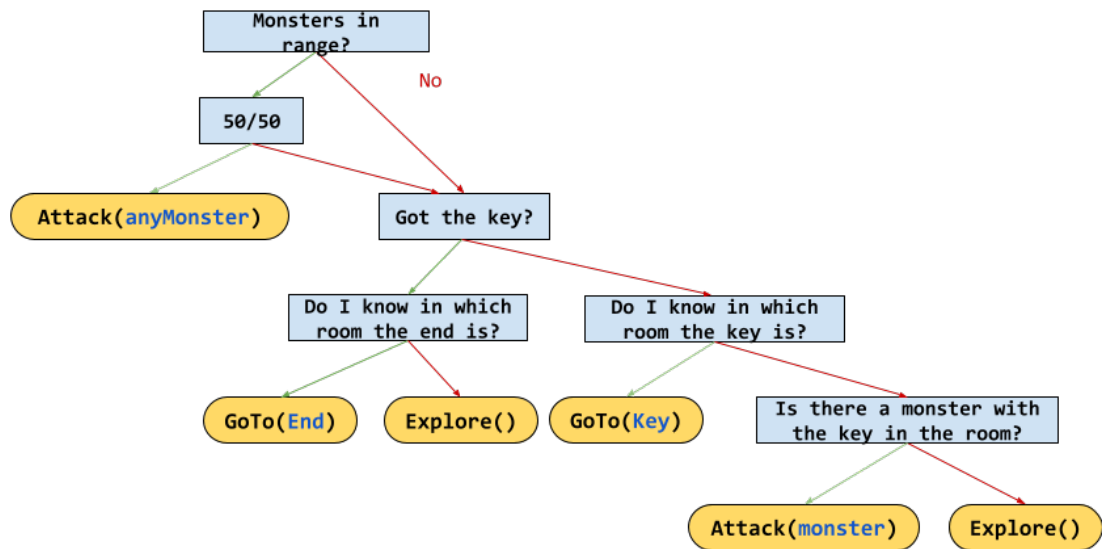


Figure 5.3: Diagram of the binary decision tree executed by the Adventurers' AI

The Adventurers' behavior had several methods implemented as actions of the Binary Decision Tree. Those are:

- ***Attack(anyMonster)*** (Attack any monster): Selects a randomly chosen monster from the reachable.

- ***GoTo(Target)*** (Go towards a target): Tries to find the path towards the target using the path finding algorithm. If the path is found but the target is out of range, the system goes backwards through it until finds a tile in the range of movement of the Adventurer. If the target is in range and available, it simply selects it as target of its action.
- ***Attack(Monster)*** (Attack a given monster): This is the equivalent to *GoTo(Target)* but using an enemy unit as target. This is defined and named like this for better understanding.
- ***Explore()*** (Looks for a Door tile to go to): Whenever an Adventurer steps on a Door, it is considered that the unit is on every room that the Door connects at the same time. This mechanic is constantly being used by the AI to quickly discover and save in the Adventurer's brain the rooms of the dungeon and its content. Whenever an Adventurer decides to «explore», tries to choose a Door which leads into an unexplored (not yet saved or «remembered») room. If it does not find any unexplored Door, it chooses one of the available doors randomly. The target Door will be the target of the *GoTo(Target)* action.

These actions were specifically designed in order to make the adventurers use the same methods of interaction with the game world as the player does. In this way, the Adventurers are independent agents working on the game world using the same exact mechanisms that the player does. It was that the AI would work like this in order to make the game scalable, as it could be interesting to develop some multiplayer features in the future.

5.1.2 Second sprint

Work done during the third and fourth weeks of work. The first version of the tutorial was made. This tutorial explained the main mechanics of the game, but did not show to the player how to use them implying the player during it, as it does the current version of the tutorial. It was also prepared the first playtest. For this, a Google Forms test was made (see Appendix C) and some game rules and bugs of the game were fixed in order to have a build done for the testers.

After that, a logo for the game was created (see Figure 5.6). During this sprint, the artistic work was started. Some concept arts for the monsters were made (see Figure 5.4-5.5) using Krita (see Section 2.2). The 2D art of the GUI currently used in the game (see Section 3.4) was also done during this and the next sprint.

Finally, the currently used 3D model for a board tile on the game was done.



Figure 5.4: Concept art for the Rat and the Spider



Figure 5.5: Concept art for the Goblin and Troll



Figure 5.6: Upside Dungeon logotype

5.1.3 Third sprint

Work done during the fifth and sixth weeks of work. A post-processing layer was created and added. Also made some other visual improvements to the tool-tip box (GUI element), adding the stats icons. Particles that show the amount of damage applied after an attack were created. Also, a system that showed the cost to step on every highlighted tile while selecting a monster was done. Finally, during this sprint, the 3D models for every unit of the game (see Figure 5.7-5.8) were done using Blender (see Section 2.2).



Figure 5.7: Detail of the Monsters' models

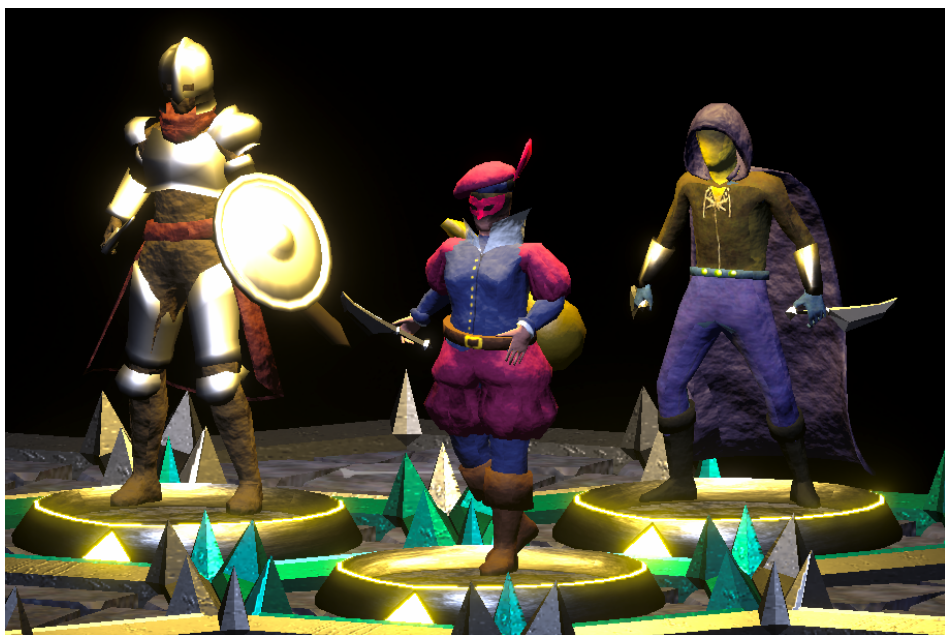


Figure 5.8: Detail of the Adventurers' models

5.1.4 Fourth sprint

Work done during the last weeks of the development of the project. Also, during this sprint, all the sound of the game was done. The OST was developed and the voices of the Adventurers and Monsters were recorded.

The OST was developed using Soundtrap (see Section 2.2) and has three themes inspired by the Dead Cells OST[4]. One main theme can be listened on the main menu of the game. There is also a theme for each side (Adventurers and Monsters). Every theme of the OST uses the same motif, with main variances on the instruments used. Adventurers' theme is brighter so it uses a Celtic Harp as main instrument. Monsters' theme should be darker and ominous so it uses a pizzicato double bass and a tuba.

The author counted with the collaboration of Alicia Montoya Peramo in order to perform the female voices of the game. Every adventurer has a total of four lines to be played randomly when being selected; four more to be played when attacking, four more for receiving damage and one death line. Monsters have four different attacking sounds, four different death and receiving damage sounds, four sounds for selection and four sounds for spawning. Every of these were recorded at the LABCOM¹. Every of them were edited and trimmed using Soundtrap. There were also done, using Soundtrap, four strike sounds and an SFX for the key when it is being picked-up.

Also, an original cursor for the game that glows when selecting a unit or spawning was done.

Finally, a set of animations was done for every unit modeled during the previous sprint. Every unit has an idle animation, an animation to play when being selected, one more to play when being attacked and a last one to play when attacking.

With all of this done, the second play test was prepared and a new version of the game launched. This version is very similar to the final one, as the technical and artistic development is done.

During the last week, the new version of the tutorial was done. Also were added the options features to change the general volume of the game and the language. Finally, in order to show that the game system is easily scalable, a second level was done (different to the tutorial level).

5.2 Results

As a result of the work, the objectives described during previous chapters have been accomplished: the development of a turn-based tactics game based on classic role-playing board games is done. The project was indeed designed to bear some kind of changes in mind. The content described in the first design of the game was an ideal case of the development of the project. The maximum level of content that the project could comprise was compounded of three levels, three different enemies, four kinds of monsters and two kinds of nests, every game agent with its respective actions. A minimum version of the project would include one single scenario/level, one single enemy character with a

¹Laboratory of communication sciences, Universitat Jaume I of Castellon



Figure 5.9: Screenshot of the final version of the game

single action and a single kind of monster. As expected, in this way, the level of content of the game has been restrained. This was about simplifying the structure of levels and agents, not about having a simpler art style or finish touch.

There were some troubles with the implementation of the AI. Its design was, at first, much more complicated than the final one. The project was supposed to focus on other aspects, so I did not have too much time to implement a complicated AI. Also, the binary decision tree was going to be implemented as a real binary node tree, but the communication between nodes required a design which there was no time for. Finally, it was decided that the whole AI would be programmed using an «if-else» structure using a single script.

Units were going to be able to perform two different actions, such as «throw a dagger» and «stab». The bard was going to be able to perform an action that would have her partners healed. This was complicating the AI design and the user interaction too much, so those game rules were finally modified.

On its first design, the game had three levels and the Adventurers had several health bars (and so much more health points). Whenever an adventurer would have lost enough HP by receiving damage, it would lose the current health bar and its level would increase in one unit and start a new health bar. The higher the adventurer's level, the higher its attributes (health and speed). If the last health bar is consumed, the adventurer would die for the rest of the party and drop the key. This was thought in order to have three different levels with an interesting difficulty curve. As expected, there was no time to spend on it, so it was decided that the gameplay time would be decreased.

The game developed is uploaded on a code repository using GitHub (see Appendix B). There is also a playable build of the game that can be accessed using the next link: <https://drive.google.com/file/d/1WsJn1qIeCxQ0UoD0tpIXtIfhVseDnEL2/view?usp=sharing>.

Of course, the game could get a better finish and more advanced features that will be explained during the next chapter.

CONCLUSIONS AND FUTURE WORK

Contents

6.1	Conclusions	53
6.2	Future work	54

6.1 Conclusions

I have always thought that the Final Degree Work was specially important in this degree. As the students get a very multidisciplinary and generalist experience during the degree, I think it is very interesting to put these in common when facing the Final Degree Work. I thought a lot about what kind of game I would do for this project and I got several different ideas during the degree for this. Finally, I went back to basics.

Almost the very first idea I had was to just take an interesting board game and bringing it into a videogame. Why would I do that? Board games are getting more and more complicated, with several tokens, marks and rules. Placing every item from the game's box on a table, spending one hundred and sixty minutes on a single game and gather the game tokens and boards up back into their box sometimes feels pretty exhausting. All of this if you have time enough to end the game. If the game gets too long, you might see yourself forced to cut it off and lose your progress.

Of course I am just talking about the issues of board games, but they have many other interesting aspects, don't blame me. What I am trying to say is that videogames can solve this kind of problems. You can play just by turning the computer on and clicking on the game icon. The game is getting too long? Do not worry, most videogames include a feature to automatically save the game state; just close the game window if you have

some other productive activities to fulfill. Also, the calculus is much quicker, figures can move in awesome animations and an AI can bear the enemies' behaviour, everything sounds cool!

Some of my friends have already enjoyed the features of *Upside Downgeon*. They want me to continue developing this game. I find the project very interesting, as I felt free to almost do whatever I wanted with it during its development. My objective was that they could have a good time playing a game that reflected my thoughts about *board-videogames*. I spent some fun (and sometimes not so fun) time developing it.

6.2 Future work

Upside Downgeon is designed to be scalable. I wish I can continue its development in the future. For now, I will probably keep its development as a hobby. I will probably work on its multiplayer capability. As explained previously, the AI works using the same methods that the user inputs does. Implementing a feature to hide the content of the hidden rooms of the board for the players bearing the adventurers and a peer-to-peer online multiplayer structure would extend a lot the capabilities of the game.

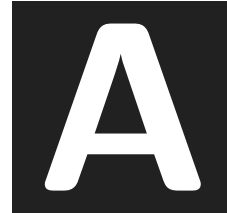
Furthermore, it would be interesting to create several new kind of monsters and adventurers, with different stats, skills and kinds of attack; more interesting boards with new types of tile, with different costs and effects... definitively, several new game rules, agents and levels.

If it seems that the work should or could be continued, it is convenient to include this section to indicate how and in which ways. It should also be indicated if the student itself plans to do so in the future.

BIBLIOGRAPHY

- [1] Electronic Arts. *Dungeon Keeper*. <https://www.ea.com/es-es/games/dungeon-keeper>.
- [2] Resolution Games. *Demeo*. <https://www.resolutiongames.com/demeo>.
- [3] Indeed.com. Salary finder. <https://es.indeed.com/career/salaries?from=gnav-title-webapp>.
- [4] Yoann Laulan. Dead cells ost. https://music.youtube.com/playlist?list=0LAK5uy_ntMfBbru_jNuI-lB5UvpUQ00TQxo8nSwE.
- [5] Wizards of the coast. *Dungeons & Dragons*. <https://dnd.wizards.com>.
- [6] Cyanide Studio. *Blood Bowl 3*. <http://www.cyanide-studio.com/blood-bowl-3>.
- [7] Unity. New input system. <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.0/manual/QuickStartGuide.html>.
- [8] Wikipedia. Dungeon crawl games. https://en.wikipedia.org/wiki/Dungeon_crawl.

APPENDIX



DIARY OF DEVELOPMENT

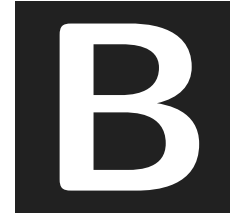
Here is the whole document of the diary of development. This chart shows the time spent in every single real task during the development of the project. This information has been daily recorded.

Tasks	Monday	Tuesday	Wednesday	Thursday	Friday
	28/03	29/03	30/03	31/03	01/04
First project planning	1 hrs 40 min	0	0	0	0
Unity3D project	10 min	0	0	0	0
Code Repository	10 min	0	0	0	0
Crating a tile 3D model	40 min	15 min	0	0	0
Board implementation	1 hrs 20 min	2hrs 45 min	1 hrs 30 min	0	0
User interaction (Input)	0	2 hrs 10 min	0	1 hrs	0
Modeling a simple character	0	20 min	0	0	0
Setting the project for HD render pipeline	0	1 hrs	0	0	0
Materials and visual feedback	0	1 hrs 30 min	30 min	0	0
Downgrading from HDRP to URP (due to bug)	0	0	1 hrs	0	0
Simple agent movement	0	0	4 hrs	7 hrs	30 min
Game rules: passing turn	0	0	0	0	3 hrs
Game rules: ending stage (winning and losing)	0	0	0	0	1 hrs
Modeling a key tile	0	0	1 hrs	0	0
Picking objects	0	0	0	0	2 hrs 30 min
User interface	0	0	0	0	1 hrs
Total	4 hrs	8 hrs	8 hrs	8 hrs	8 hrs
Tasks	Monday	Tuesday	Wednesday	Thursday	Friday
	04/04	05/04	06/04	07/04	08/04
Spawning units using UI	1 hrs	0	0	0	0
Debugging pathfinding	1 hrs 30 min	0	30 min	0	0
Mana costs and turns system (game rules)	1 hrs 30 min	0	0	0	0
Tooltip to show unit's stats (UI)	0	2 hrs 30 min	0	0	0
Working on UI colors for better contrast	0	30 min	0	0	0
Vertical fog	0	1 hrs 30 min	0	0	0
Moving the camera	0	3 hrs	0	0	0
New board features (can not spawn anywhere)	0	0	2 hrs 30 min	0	0
New agents features (Adding actions and abilities)	0	0	1 hrs 30 min	0	0
Artificial Intelligence for the Adventurers	0	0	3 hrs 30 min	8 hrs	5 hrs
Concept art	0	0	0	0	3 hrs
Total	4 hrs	8 hrs	8 hrs	8 hrs	8 hrs
Tasks	Monday	Tuesday	Wednesday	Thursday	Friday
	11/04	12/04	13/04	14/04	15/04
Concept art	4 hrs	3 hrs	0	0	0
Artificial Intelligence for the Adventurers	0	5 hrs	3 hrs	0	0
User Interaction	0	0	5 hrs	0	0
Total	4 hrs	8 hrs	8 hrs	0	0
Tasks	Monday	Tuesday	Wednesday	Thursday	Friday
	25/04	26/04	27/04	28/04	29/04
Artificial Intelligence for the Adventurers	3 hrs	0	0	0	0
Debugging	1 hrs	2 hrs	0	0	0
Fixing stats, costs and game rules	0	1 hrs	0	0	0
UI (menu)	0	1 hrs	0	0	0
Tutorial	0	4 hrs	0	0	0
Pause menu and user interaction	0	0	2 hrs	0	0
Polishing and build (For the alpha test)	0	0	30 min	0	0
Google form test for the alpha testing	0	0	1 hrs 30 min	0	0
New 3D asset for the tile (Modeling)	0	0	4 hrs	0	0
Game title, logo and main menu art (2D art)	0	0	0	8 hrs	0
UI (In game HUD 2D art)	0	0	0	0	7hrs 30 min
Fixing playtest	0	0	0	0	30 min
Total	4 hrs	8 hrs	4 hrs	8 hrs	8 hrs

Tasks	Monday	Tuesday	Wednesday	Thursday	Friday
	02/05	03/05	04/05	05/05	06/05
Adding new models and materials	1 hrs	0	0	0	0
Post-Processing	1 hrs	0	0	0	0
Stats Icons (UI 2D art)	2 hrs	0	0	0	0
Tooltip box Stats icons and animation	0	8 hrs	0	0	0
Showing damage and effects	0	0	4 hrs 30 min	0	0
Showing tiles cost	0	0	3 hrs	0	0
Total	4 hrs	8 hrs	8 hrs	0	0
Tasks	Monday	Tuesday	Wednesday	Thursday	Friday
	09/05	10/05	11/05	12/05	13/05
Bug fixes	2 hrs 30 min	0	0	0	0
Low Poly character, bones and animation (Rogue)	1 hrs 30 min	0	0	0	0
Low Poly character, bones and animation (Rogue)	0	3 hrs 30 min	0	0	0
Low Poly character, bones and animation (Rogue)	0	4 hrs 30 min	0	0	0
Low Poly character, bones and animation (Warrior)	0	0	4 hrs	0	0
Low Poly character, bones and animation (Bard)	0	0	4 hrs	0	0
Low Poly character, bones and animation (Troll)	0	0	0	4 hrs	0
Low Poly character, bones and animation (Goblin)	0	0	0	4 hrs	0
Low Poly character, bones and animation (Rat)	0	0	0	0	4 hrs
3D model optimisations (on Troll and Goblin)	0	0	0	0	1 hrs 30 min
Game balance	0	0	0	0	30 min
Bug fixing (AI)	0	0	0	0	2 hrs
Total	4 hrs	8 hrs	8 hrs	8 hrs	8 hrs
Tasks	Monday	Tuesday	Wednesday	Thursday	Friday
	16/05	17/05	18/05	19/05	20/05
Personalised cursor	2 hrs	0	0	0	0
Writing voice lines	30 min	0	0	0	0
Recording voices (LABCOM)	3 hrs	0	0	0	0
Trimming and organising recorded audio	2 hrs 30 min	3 hrs 30 min	0	0	0
Adding voices to the game	0	4 hrs 30 min	0	0	0
Music - Main theme (Soundtrap)	0	0	3 hrs 30 min	0	0
Music - Monsters' and adventurers' themes (Soundtrap)	0	0	3 hrs	0	0
SFX (Soundtrap)	0	0	1 hrs 30 min	0	0
Characters' animation sets (Adventurers)	0	0	0	5 hrs 30 min	0
Characters' animation sets (Monsters)	0	0	0	2 hrs 30 min	3 hrs
Game events and animation coordination	0	0	0	0	1 hrs
Polishing and build (For the alpha test)	0	0	0	0	3 hrs 30 min
Google form test for the alpha testing	0	0	0	0	30 min
Total	8 hrs	8 hrs	8 hrs	8 hrs	8 hrs
Tasks	Monday	Tuesday	Wednesday	Thursday	Friday
	23/05	24/05	25/05	26/05	27/05
Project Memory	3 hrs	0	0	8 hrs	4 hrs
Basic Tutorial	5 hrs	0	0	0	0
Movement Tutorial	0	4 hrs	0	0	0
Localising the game (Spanish)	0	4 hrs	2 hrs	0	0
Adding audio volume settings	0	0	2 hrs 30 min	0	0
Adding player preferences saving	0	0	30 min	0	0
Adding a new level/board to the game	0	0	3 hrs	0	0
Total	8 hrs	8 hrs	8 hrs	8 hrs	4 hrs
Tasks	Monday	Tuesday	Wednesday	Thursday	Friday
	06/06	07/06	08/06	09/06	10/06
Project Memory	8 hrs	8 hrs	8 hrs	8 hrs	8 hrs
Total	8 hrs	8 hrs	8 hrs	8 hrs	8 hrs

A.1 Bibliography

Javier Albors Caño. Diary of Development. <https://docs.google.com/spreadsheets/d/1cag0R0hMM2HI4AXSEvrg-8YU725LM0pGdSAGsd-xSi0/edit?usp=sharing>



SOURCE CODE AND OTHER CONSIDERATIONS

The videogame developed for this Final Work Degree is composed by several scripts. I used GitHub as a tool of version control, so the Unity project and every script is uploaded to a code repository that can be checked using the next link:

<https://github.com/javiwagh/UpsideDowngeon-TFG>.

Also, a playable build of the videogame can be downloaded using the next link:

<https://drive.google.com/file/d/1WsJn1qIeCxQ0UoD0tpIXtIfhVseDnEL2/view?usp=sharing>.

Finally, there have been made two videos using the build that can be downloaded using the previous link. One of them is a trailer used to present the videogame. The other is a commented *gameplay* that shows the flow of the game. Check them using the next link:

- Trailer: https://drive.google.com/file/d/1JZEbuKTqGhPEk0U5d_oe_iaBalRss43k/view?usp=sharing.
- Gameplay: <https://drive.google.com/file/d/1QnqUTug4vHSBww0YVQC0L4Ie2XiRDrGS/view?usp=sharing>.



PLAY TESTING

Dugning the project, two play tests were made in order to show the videogame to possible real players. By coincidence, these all were twenty to thirty years-old-men. To check the tests and the builds used on them, check the next links:

<https://forms.gle/w3BnQU34mZzVw3ys7>.

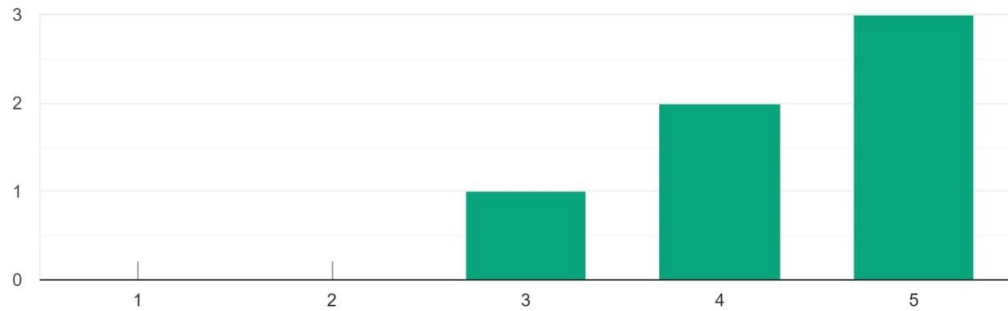
<https://forms.gle/iBEHKUee4KR1aWxP6>.

To consult the answers of the second playtest, the next link can be checked:

<https://docs.google.com/spreadsheets/d/1GP7KFfGXYpqj7YYPmzPh9UZJQvA0Bvho0euNh30FBuQ/edit?usp=sharing>.

Valora en una escala del 1 al 5 la calidad del aspecto de los menús y el interfaz de usuario .

6 respuestas



¿Qué aspectos de los diversos menús e interfaces te han incomodado más? Piensa en qué podría mejorarse. 6 respuestas

todo se puede mejorar pero no le e visto fallo menos las partes en ingles

La confirmación con dos clicks me sigue rallando a veces, pero todo está muy fino

Se me ha "bugeao" una poca, pero taba bonito, me agrada.

Lo único que añadiría sería un menú de opciones en el menú principal para poder regular cosas como el volumen o la resolución pero por el resto es muy agradable y creativo.

los textos costaban un poco de ver

Echo de menos poder sacar la interfaz pulsando un botón

¿Qué aspectos de los diversos menús e interfaces te han parecido más interesantes? 6

respuestas

todos

Es difícil hacer que el tutorial no dé asco, grats. Los textos, el tablero y los iconos son muy friendly, no dan miedo como en otros juegos de este estilo.

Es simple y entendible.

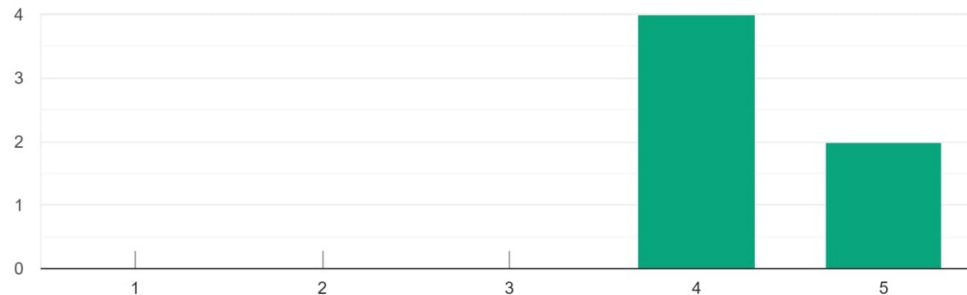
En general el titulo del juego que se muestra en el menú principal. Me gusta mucho esa fuente/colocación de las letras junto con los detalles que esta posee.

las estadísticas que se mostraban al seleccionar

las animaciones del menú. Están logradas.

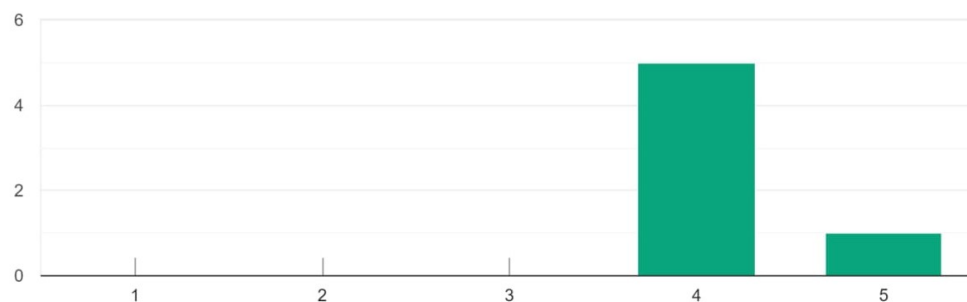
Valora en una escala del 1 al 5 la calidad de los modelos de los elementos del tablero (monstruos, aventureros, casillas...).

6 respuestas



Valora en una escala del 1 al 5 la calidad de las animaciones de los elementos del tablero.

6 respuestas



¿Qué aspectos de los gráficos del juego te han incomodado más? Piensa en qué podría mejorarse. 6 respuestas

si pienso que si pero con los recursos que a tenido el desarrollador lo veo muy bien

Bastante graciosos los monstruos, algo más planos los héroes, les falta que parezcan más arrastraos xd.

QUIERO PUERTAS DE VERDAD

La rata es mu fea mu fea

Los heroes un poco sosetes, pero tan bien, cumplen su rol

Se podría mejorar los efectos al golpear de una tropa: Algo de sangre para las armas blancas, escombros o polvo para los golpes del troll, partículas para el veneno o la parálisis. Quizás algo mas de decoración para las baldosas o las paredes del la mazmorra pero eso ya es pura estética.

el menú principal no se acoplaba bien

A veces se nota un poco raro el estilo, pero es porque no estoy hecho a ello.

¿Qué aspectos de los gráficos del juego te han resultado más atractivos? 6 respuestas

los bichos

La rata es maravillosa. En general, pasar de hexágonos feos a esto es un buen glowup

La araña ta bonita, el gitanopayo es clave, y el golem es mi amigo

La animacion de muerte del gitanopayo es mu graciosa, me agrada

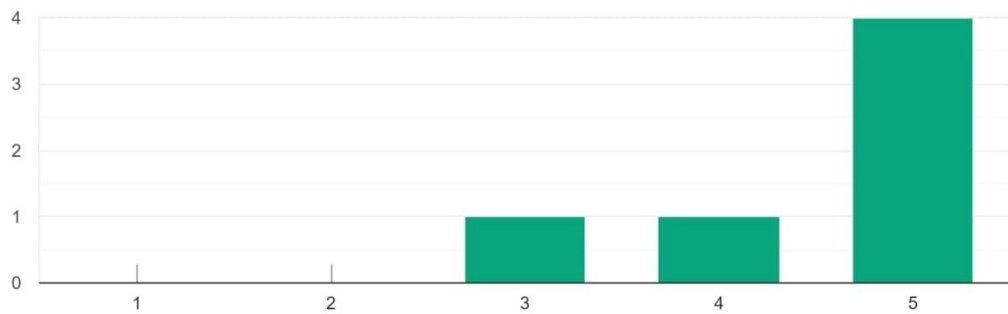
Me ha sorprendido todo en general. De la primera alfa a esta ha habido cambios muy muy grandes. No paraba de dar pequeños brincos con cada modelo de los bichillos o de los aventureros. Estaban muy bien.

los muñecos

El tablero en cuadrículas hexagonales. Me ha flipado.

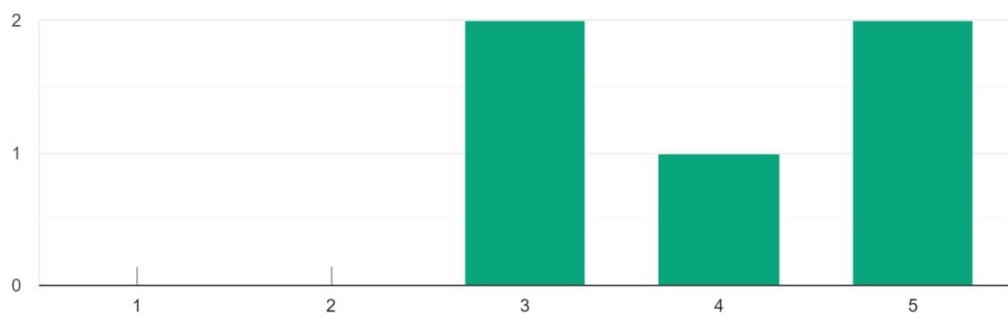
Valora en una escala del 1 al 5 la calidad de la música del juego.

6 respuestas



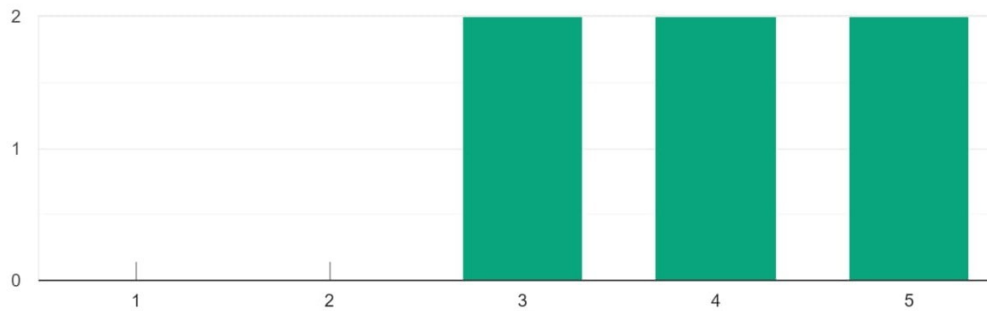
Valora en una escala del 1 al 5 la calidad de los SFX del juego.

5 respuestas



Valora en una escala del 1 al 5 la calidad del doblaje.

6 respuestas



¿Qué aspectos de la sonorización del juego te han incomodado más? Piensa en qué podría mejorarse. 6 respuestas

esta muy bien

Los monstruos gruñen bajo, pero a los héroes los escuchaba como bajito raro y no les entendía bien.

Las voces de los heroes se escuchaban una poquita bajas, pero tiene potencial

Distintas tracks de sonido para los niveles estarían bien pero no es muy importante. Lo que hay ahora esta bastante bien.

las frases del bardo

Ninguno. Me ha molado mucho.

¿Qué aspectos de la sonorización del juego te han parecido más atractivos? 6 respuestas

los doblajes

Los ruidacos de los bichos. La música está bien pero pasa bastante desapercibida.

La musica de ambiente, aunque como detalle (si no es mucho pedir) estaria lindo que conforme se desarrolla el combate vaya evolucionando un poco

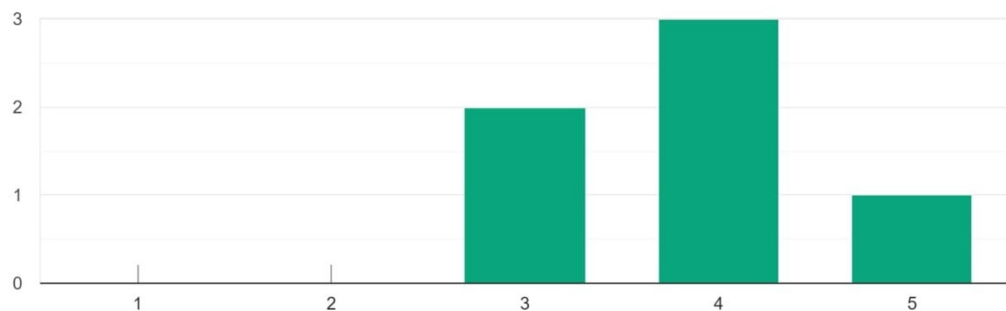
Lo sonidos de los monstruos y el los aventureros. No me esperaba que estuvieran doblados o que tuvieran soniditos al golpear y al moverse. Me ha gustado mucho ese detalle sobre todo con los goblins.

los ruidos de los monstruos

Uff, que parezca un juego de principios de los 2000. Tiene ese toque nostálgico que me encanta

Valora en una escala del 1 al 5 la naturalidad de los enemigos (Inteligencia Artificial). ¿Te parecía que sus decisiones eran razonables?

6 respuestas



¿Ha habido algún comportamiento especialmente extraño? Por ejemplo, las piezas no se movían o llevaban a cabo movimientos en bucle. Por favor, trata de describirlo. 6 respuestas

se mueben un poco arreu los exploradores

Tras descubrir a mis bichos en la sala de la izquierda, el héroe se ha ido hacia la derecha para volver hacia la izquierda como a suicidarse xd

Había veces que a mi percepción había movimientos mas óptimos y la ia se movía raro, pero en lo general está bien.

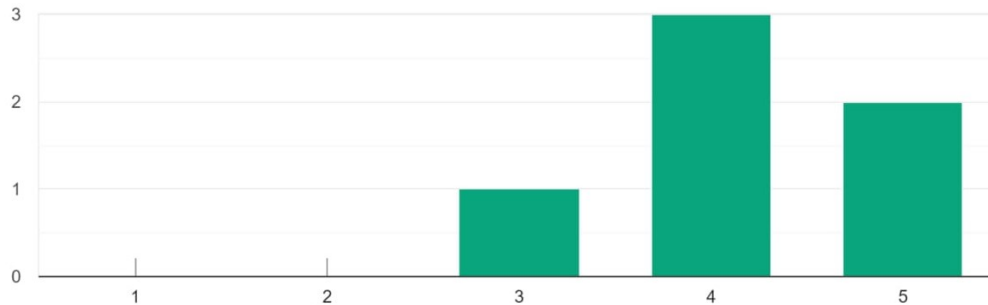
En ningún momento he visto un comportamiento extraño en las figuras. Su movimiento a mi me resulta algo natural debido a que se mueven en busca de la llave y una vez que la ubican pues van hacia allá para tomarla y escapar o proceder en la mazmorra. Debido a esa base me parece normal y acertado que al principio se dividan y cada uno vaya a donde le de la gana.

aveces el hacian caminos raros para llegar al sitio

La verdad es que no

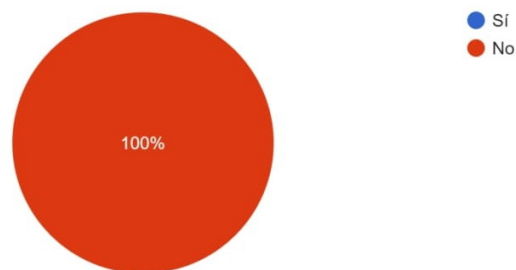
¿Cómo has sentido el juego en general? En una escala del 1 al 5, ¿cuán intuitiva te ha resultado la experiencia?

6 respuestas



¿Has experimentado algún bug crítico (Error que no te haya permitido continuar jugando)?

6 respuestas



Si has experimentado un bug crítico, por favor, trata de describirlo. 2 respuestas

Ninguno uwu

nada

¿Crees haber experimentado otro algún fallo o error menos importante? ¿Puedes describirlo? 1 respuesta

Por la resolución de mi pantalla se bugueó el "cliente", pero mas o menos lo pude solucionar.

Por favor, describe cuáles han sido los aspectos del juego que más te han incomodado. Puedes valorar la interacción con el usuario, el diseño del interfaz, el tutorial, diseño de juego y mecánicas, etc. Puedes valorar tanto aspectos técnicos como artísticos. 6 respuestas

ninguna

Me falta un zoom, le he dado un par de veces a la rueda del ratón queriendo acercarme al goblin uf. También estaría bien la típica opción de ir pasando de unidad con las flechas del teclado por ejemplo, para ver que no te dejas ningún monstruo sin mover y no tener que ir buscándolos por el mapa

Nada en concreto, el diseño de la rata fea como mucho jajaja

A ver. He visto una gran mejora respecto a la anterior alfa. La jugabilidad a mejorado enormemente, el tema gráficos también y podríamos decir que todo ha mejorado. Desde mi punto de vista quizás mejoraría el target o el ataque de los monstruos de cara a los que tiene la habilidad de "backstab". No se como explicarlo bien pero como poder orientar tu al monstruo o poder indicarle el path a seguir y después definir el ataque por ese ángulo. Lo digo porque en una run que he jugado solo ha goblins pues a la hora de aprovechar la habilidad estos no lo hacían del todo con el primero golpe y simplemente tiraban hacia delante por la vía mas rápida. Ahí me hubiera gustado poder definirle la distancia de movimiento y el lado por donde atacar (Siempre y cuando tenga la suficiente)

los colores de los menus

mándamelo cuando haya tuto, que necesito entender como funcionan las casillas de distintos colores. En general el resto está bastante bien. Me ha gustado mucho el juego.

¿Qué aspectos de la experiencia en general te han parecido más interesantes? ¿Qué partes del diseño crees que podrían potenciarse? Puedes valorar tanto aspectos técnicos como artísticos.⁶ respuestas

porfavor mas niveles

Como tutorial, un 10. Se integra con el gameplay y no se hace pesado y, de primeras, no es excesivamente fácil (la primera vez que lo jugué en modo poligonal me mataban los bichos xd).

Me ha agradado invocar un ejercito de gitanopayos, muy meme

Esta experiencia ha sido fantástica. Respecto a la anterior alfa he visto muchos cambios que me han gustado mucho y que se han tenido en cuenta en las reseñas (Como el conservar parte del "mana" si no lo gastas todo o que los monstruos también tengan dos movimientos para no estar en una desventaja tan grande frente a los aventureros). Las animaciones, los diseños y el apartado sonoro del juego me han encantado (Sobretudo las miniaturas de los monstruillos. Son muy tiernas. Las arañas son geniales).

Pienso que se puede explotar mucho esta idea de ser el "amo de la mazmorra" añadiendo mas monstruos por niveles del juego y dándole algo de interactividad al terreno, como dije en la anterior reseña, pero esto es de cara a un futuro largo, claro. Por ahora, lo que he jugado, me ha agradado bastante y ha logrado conseguir el cometido para el que se crearon los videojuegos: Divertir los jugadores y consumidores de estos.

era muy placentero de ver

Todo. Es un concepto atractivo.