



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

Prototype of a video game for computer using Unity

Bachelor's Thesis

Bachelor's Degree in Video Game Design and Development

Latorre Freixes, Marc
Plan: 2014

Zúñiga Zarate, Ana Gabriela

Index

Summary	7
Key Words	8
Links	8
Tables Index	9
Figures index	10
Glossary	13
1. Introduction	16
1.1 Motivation	16
1.2 Problem formulation	16
1.3 General objectives	16
1.4 Specific objectives	16
1.5 Scope of the project	17
2. State of the art	18
2.1 Technology	20
2.1.1 Unity	20
2.1.2 Unreal Engine	21
2.2 Market research study	21
3. Project management	26
3.1 Procedures and tools for the tracking of the project	26
3.1.1 GANTT	26
3.1.2 Personal timetable	28
3.1.3 Hack N Plan	28
3.1.4 Bitbucket repository	29
3.2 Evaluation tools	29
3.2.1 Personal and close people testing's	29
3.2.2 Playtesting between target audience	29
3.3 SWOT	30
3.4 Risks and contingency plan	31
3.5 Initial costs analysis	31
3.6 Initial planification modification (05/04/2019)	38
3.6.1 HP bars on the enemies	38
3.6.2 Missing map sprites	38
3.6.3 Missing particle effects	38
3.6.4 Player respawning system	38

3.6.5	Checkpoints	38
3.6.6	Scene management	39
3.6.7	Maps interactions	39
3.6.8	Assets remodeling	39
3.6.9	Balancing	40
3.6.10	New GANTT	40
4.	Methodology	42
4.1	Research	42
4.2	Design	42
4.3	Development	43
4.4	Testing	43
5.	Project development	44
5.1	Assets research and selection	44
5.2	Design	44
5.2.1	Game design	44
5.2.1.1	Game pillars	44
5.2.1.1.1	Gameplay	44
5.2.1.1.2	Mobility	44
5.2.1.1.3	Fast	44
5.2.1.2	Game story	45
5.2.1.3	Flowchart	45
5.2.2	Characters design	46
5.2.2.1	Combat attributes example	47
5.2.3	Player design	47
5.2.4	Enemies design	48
5.2.4.1	Enemy bosses	49
5.2.4.1.1	Monk	49
5.2.4.1.2	Ice Gorilla	49
5.2.4.1.3	Demon	50
5.2.4.2	Regular enemies	51
5.2.4.2.1	Ice Hedgehog	51
5.2.4.2.2	Archer	52
5.2.4.2.3	Dwarf	52
5.2.4.2.4	Bird	52
5.2.4.2.5	Lizard monster	53

5.2.4.2.6 Fluffy monster	53
5.2.4.2.7 Imp	54
5.2.4.2.8 One-Eyed Imp	54
5.2.4.2.9 Little Yeti	54
5.2.5 Items design	55
5.2.5.1 Collectable items	55
5.2.5.2 Items on the Store	55
5.2.6 Levels design	55
5.2.6.1 Level 1	57
5.2.6.2 Boss Level 1	59
5.2.6.3 Level 2	60
5.2.6.4 Boss Level 2	64
5.2.6.5 Level 3	65
5.2.6.6 Boss Level 3	67
5.2.7 User interface design	68
5.2.7.1 HUD	68
5.2.7.1.1 Skills icons	69
5.2.7.2 Store	70
5.2.7.3 Character panel	70
5.2.7.4 Menu panel	71
5.2.8 Audio and sound effects	71
5.2.9 Levels tile map sprite sheets	71
5.2.9.1 Jungle tilemap	71
5.2.9.2 Dark theme tilemap	72
5.2.9.3 Snow tilemap	72
5.2.9.4 Levels on Unity	73
5.3 Programming	74
5.3.1 Player	74
5.3.1.1 Player respawning	75
5.3.2 Enemies	75
5.3.3 Item system	75
5.3.4 Levels	76
5.3.5 User Interface	76
5.3.5.1 HUD	77
5.3.5.2 Character panel	77
5.3.5.3 Store	77

5.3.5.4 Settings	77
5.3.6 Scene management	77
5.4 Testing and bugs	78
5.4.1 Personal and close people playtesting's	79
5.4.2 Post personal and close people testing bugs	79
5.4.3 Post personal and close people testing changes	80
5.4.3.1 Little Yeti and Ice Gorilla design	80
5.4.3.2 Level design change and new sprites	81
5.4.3.3 New level 2 zone	81
5.4.3.4 New particle effects	82
5.4.3.4.1 Player buffs activation	82
5.4.3.4.2 Teleport	83
5.4.3.4.3 Summon enemies	84
5.4.3.4.4 Ice lances and fire balls spawners	84
5.4.3.4.5 Dark Shadows skill	84
5.4.3.4.6 Player respawn	85
5.4.3.5 New audio effects	85
5.4.3.6 Player guidance	86
5.4.3.7 Player buffs design	86
5.4.3.8 Second boss level redesign	86
5.4.4 First external playtesting	87
5.4.5 First external playtesting results	88
5.4.6 First external playtesting evaluation	91
5.4.7 First external playtesting changes	91
5.4.7.1 Controls redesign	91
5.4.7.2 Damage popup	92
5.4.7.3 Skill points and store feedback	92
5.4.7.4 Change theme each level	93
5.4.7.5 Armor redesign	93
5.4.7.6 First boss level redesign	94
5.4.7.7 Buffs reset on player death	94
5.4.7.8 Other minor changes	94
5.4.8 Second external playtesting	94
5.4.9 Second external playtesting results	95
5.4.10 Second external playtesting evaluation	97
5.4.11 Second external playtesting changes	97

5.4.11.1 UI buttons changes	97
5.4.11.2 Level 3 checkpoint	97
5.4.11.3 Fluffy monster wick spark	97
5.4.11.4 Skills description	98
5.4.12 Unity bug	98
5.4.13 Soak idle testing	98
5.5 Polish and balancing	98
5.5.1 First balancing	99
5.5.1.1 Player stats	99
5.5.1.2 Store items	100
5.5.1.3 Player experience	101
5.5.1.4 Player potential damage and defense	102
5.5.1.5 Enemies balance	103
5.5.1.5.1 Bosses balance	105
5.5.2 Balancing after first external playtesting	105
5.5.3 Third level balancing	106
5.5.3.1 Enemies balance (3 rd level)	106
5.5.3.2 Boss balance (3 rd level)	107
5.5.4 Balancing after second external playtesting	108
5.5.5 Final balancing	109
5.5.6 Polish	109
5.5.6.1 Demon boss and its level redesign	109
5.5.6.2 Tutorial level	109
5.5.6.3 Spawners sprite change	109
5.5.6.4 Typography change	110
5.5.6.5 Elastic spring sprite	110
5.5.6.6 UI aesthetic redesign	110
5.5.6.7 Door animation	112
5.5.6.8 Boss levels closing door	112
5.5.6.9 Level 3 new platform	112
5.5.6.10 Main menu music	112
5.5.6.11 Victory screen	112
5.5.6.12 Main menu introduction	113
5.5.6.13 Bugfixes	113
5.6 Website	114
6. Conclusions and future work	115

6.1 Project times	115
6.2 Planned objectives	115
6.2.1 General objective	115
6.2.2 Specific objectives	116
6.2.2.1 Designing the whole game	116
6.2.2.2 Manage all the tasks satisfactorily	116
6.2.2.3 Achieve a fun gameplay	116
6.2.2.4 Balance the game	116
6.2.2.5 Implement a good UI system	116
6.2.2.6 Carry out a useful playtesting	117
6.3 Future work	117
6.3.1 Player skills	117
6.3.2 Crouching	117
6.3.3 Buy health potions	117
6.3.4 List of bought items or inventory	118
6.3.5 Level score and achievements	118
6.3.6 Save and load	118
6.3.7 Difficulty levels	118
7. Webography	119

Summary

This document is about the game design and development of a prototype 2D, RPG, action platformer videogame using the game engine Unity. Creating a full videogame in those few months and being only myself was impossible, for this reason it ended up being a prototype that can be extended to a full videogame later.

The design and programming of the prototype was entirely made by me; however, the art and the music, except for some sprites and animations, was bought or gotten from free open sources.

Its primary goal was to create something fun to play for users and in order to do that, an agile methodology was followed, thanks to this a lot of problems and improvements could be done, and, in the end, the prototype actually became fun for most of the testers that played the game.

In this document it is described the initial design and all the problems, solutions and changes that it had.

The final result can even be considered more than a prototype because of the number of features and the polish it has and can be seen in [this link](#) and is under the [GNU \(General Public License\)](#) license.

Key Words

Video game, Programming, C#, Unity, Game, Development, Code, Design, 2D.

Links

Final prototype release:

https://drive.google.com/open?id=1qWJzeEXoYoFRNSjaObjY4-2_A0uTfLLy

Second External Playtesting release link:

https://drive.google.com/open?id=1Qs0_2p5N2gF0J_cWVV99GpE_AjtH3mal

First External Playtesting release link:

https://drive.google.com/open?id=1_MTaTr4O6rNBa8_tceZoO24XI9wOUTTD

Testing release 0.1 Link:

<https://drive.google.com/open?id=1U3emBg3MScRw45VegkB-mflnSZ9vAYmt>

Tables Index

Table 1: Similar games basic information.....	22
Table 2: Similar games score	23
Table 3: Similar games development information	24
Table 4: GANTT 1.....	26
Table 5: GANTT 2.....	27
Table 6: SWOT	30
Table 7: Possible risks.....	31
Table 8: Net profit best case scenario	34
Table 9: Net profit standard case scenario.....	34
Table 10: Net profit worst case scenario.....	34
Table 11: Similar games market price	35
Table 12: Estimated income	36
Table 13: Estimated costs.....	36
Table 14: Hardware depreciation.....	37
Table 15: Estimated office costs.....	37
Table 16: New GANTT 1	40
Table 17: New GANTT 2	41
Table 18: Bugs list.....	79
Table 19: Personal and close people post-testing's list.....	80
Table 20: First external playtesting (Age).....	89
Table 21: First external playtesting (Difficulty).....	90
Table 22: Controls inputs	91
Table 23: Second external playtesting (Age)	95
Table 24: Second external playtesting (Difficulty)	96
Table 25: Player initial stats (first balance; self-made).....	99
Table 26: Player scaling stats first balance (self-made).....	100
Table 27: Player skills damage (first balance; self-made).....	100
Table 28: Items list (first balance; self-made).....	101
Table 29: Level 1 enemies' stats (first balance; self-made).....	103
Table 30: Level 2 enemies' stats (first balance; part 1; self-made).....	104
Table 31: Level 2 enemies' stats (first balance; part 2; self-made).....	104
Table 32: Monk boss stats (self-made).....	105
Table 33: Ice Gorilla boss stats (first balance; self-made)	105
Table 34: Balancing (post 2nd playtesting).....	108

Figures index

Figure 1: Global games market (font: White book DEV 2018).....	18
Figure 2: Top video game companies by revenue (font: White book DEV Spain 2018).....	18
Figure 3: Territorial distribution of game companies in Spain (font: White book Spain DEV 2018).....	19
Figure 4: Software used to develop videogames by companies in Spain (font: White book Spain DEV 2018).....	20
Figure 5: Hack N Plan task example (self-made)	28
Figure 6: Best case scenario net results (self-made)	32
Figure 7: Standard case scenario net result (self-made)	32
Figure 8: Worst case scenario net result (self-made).....	33
Figure 9: Methodology (self-made).....	42
Figure 10: Flowchart (self-made)	45
Figure 11: Character panel info design (self-made).....	46
Figure 12: Monk (bough asset).....	49
Figure 13: Ice Gorilla (bough asset)	50
Figure 14: Demon (bough asset)	50
Figure 15: Ice Hedgehog (self-made).....	51
Figure 16: Hedgehog (bough asset).....	51
Figure 17: Archer (bough asset)	52
Figure 18: Dwarf (bough asset)	52
Figure 19: Bird (bough asset)	52
Figure 20: Lizard monster (bough asset)	53
Figure 21: Fluffy monster (self-made)	53
Figure 22: Fluffy monster (bought asset)	53
Figure 23: Imp (bough asset).....	54
Figure 24: One Eyed Imp (bough asset).....	54
Figure 25: Little Yeti (bough asset).....	54
Figure 26: Level 1 Part 1 (self-made).....	57
Figure 27: Level 1 Part 2 (self-made).....	57
Figure 28: Level 1 Part 3 (self-made).....	58
Figure 29: Level 1 Part 4 (self-made).....	58
Figure 30: Boss level 1 (self-made):.....	59
Figure 31: Level 2 Part 1 (self-made).....	60
Figure 32: Level 2 Part 2 (self-made).....	61
Figure 33: Level 2 Part 3 (self-made).....	61
Figure 34: Level 2 Part 4 (self-made).....	62
Figure 35: Level 2 Part 5 (self-made).....	63
Figure 36: Boss level 2 (self-made).....	64
Figure 37: Level 3 Part 1 (self-made).....	65
Figure 38: Level 3 Part 2 (self-made).....	65
Figure 39: Level 3 Part 3 (self-made).....	66
Figure 40: Level 3 Part 4 (self-made).....	66
Figure 41: Boss level 3 (self-made).....	67
Figure 42: HUD idea (self-made)	68
Figure 43: HUD definitive (self-made).....	68
Figure 44: Dash icon (bought asset)	69

Figure 45: Dagger throw icon (bought asset)	69
Figure 46: Dark shadows icon (bought asset)	69
Figure 47: Sharp sword icon (bought asset)	69
Figure 48: Extra armor icon (bought asset)	69
Figure 49: Blood sword icon (bought asset)	69
Figure 50: Deflection icon (bought asset).....	69
Figure 51: Store design (self-made).....	70
Figure 52: Character panel design (self-made).....	70
Figure 53: Jungle tilemap sprite sheet (self-made from bought assets)	71
Figure 54: Dark theme tilemap sprite sheet (self-made from bought assets)	72
Figure 55: Snow tilemap sprite sheet (self-made from bought assets)	72
Figure 56: Level 1 on Unity (self-made).....	73
Figure 57: Level 2 on Unity (self-made).....	73
Figure 58: Level 3 on Unity (self-made).....	73
Figure 59: Level 3 secret zone on Unity (self-made).....	74
Figure 60: Rock with spikes sprite (self-made)	81
Figure 61: Platform with spikes sprite 1 (self-made)	81
Figure 62: Platform with spikes sprite 2 (self-made)	81
Figure 63: Floor with spikes sprite (self-made)	81
Figure 64: Secret coins zone level 2 (self-made)	82
Figure 65: Secret teleport zone level 2 (self-made).....	82
Figure 66: Reflect damage skill effect.....	83
Figure 67: Increase attack skill effect	83
Figure 68: Increase defense skill effect	83
Figure 69: Blood sword skill effect	83
Figure 70: Teleporter effect	83
Figure 71: Summoning effect	84
Figure 72: Ice lance spawner effect.....	84
Figure 73: Fire ball spawner effect	84
Figure 74: Dark shadows skill effect	85
Figure 75: Player respawn effect.....	85
Figure 76: Direction sign sprite (bought asset).....	86
Figure 77: Danger sign sprite (bought asset).....	86
Figure 78: Level boss 02 redesign (self-made).....	87
Figure 79: Damage popup	92
Figure 80: UI alert sprite (bought asset)	92
Figure 81: Dark theme spikes platform 1 (self-made)	93
Figure 82: Dark theme spikes platform 2 (self-made)	93
Figure 83: Armor to damage reduction % scale (self-made)	94
Figure 84: Throw dagger new icon (self-made from bought asset)	97
Figure 85: Skill description popup	98
Figure 86: Maximum, average and minimum player experience (first balance; self-made).....	101
Figure 87: Potential player DPS and explosive damage (first balance; self-made)	102
Figure 88: Potential player physic and magic defense (first balance; self-made)	102
Figure 89: Level 3 enemies' stats part 1 (self-made)	106
Figure 90: Level 3 enemies' stats part 2(self-made)	107
Figure 91: Level 3 enemies' stats part 3(self-made)	107
Figure 92: Demon boss stats (self-made)	108

Figure 93: Final ice lance spawner (modified sprite)110

Figure 94: Final fire ball spawner (modified sprite)110

Figure 95: Arial (left) vs Lucida Grande Bold (right) comparison (self-made).....110

Figure 96: Elastic spring sprite sheet (self-made, from bought sprite)110

Figure 97: Main menu (before)111

Figure 98: Main menu (after)111

Figure 99: Character info (before)111

Figure 100: Character info (after).....111

Figure 101: Store (before)111

Figure 102: Store (after)111

Figure 103: Settings (before)111

Figure 104: Settings (after)111

Figure 105: Door animation (modified sprite)112

Figure 106: Boss closing door (modified sprite)112

Figure 107: Victory screen (self-made).....113

Figure 108: Main menu cover (self-made)113

Figure 109: Estimated and real hours.....115

Glossary

2D Game: Everything in the game is happening in a 2D plane.

Action Game: Games where most of the time you are fighting against enemies.

AI (Artificial Intelligence): Attempt to make the NPC's (enemies in our case) act like a human.

AI navigation tool: A tool that helps the AI to follow a particular path without colliding with obstacles and, usually, finding the best route to get to the objective.

Asset: Everything that can be used in the video game. E.g.: an animation, a sound, ...

Balanced: Not too hard neither too easy, so the player does not get stressed or bored.

Beta version: A version of a software that is functional but is used for testing with a limited number of users not related with the developers.

Boss: An enemy that is way more powerful than a regular enemy and has distinctive skills that are unique in the game.

Bug: An error on the game that leads to a wrong behavior of something or anything that is not how the game should be. E.g.: an enemy going through a wall.

Casting time: The time that passes between the start a skill is used and the time it starts doing something.

Child game object: a child game object is a game object that inherits the transformations in the world from his parent game object.

CPU (Central Processing Unit): Part of the computer that executes the instructions. The faster the CPU, the faster the computer.

Cooldown: The time a skill can be used again after just using it. E.g.: If the player uses a skill that has a 1 second cooldown, once he uses this skill, it cannot be used again after a second has passed.

Development language: Programming language used in the project.

DPS (Damage per Second): The average amount of damage that a player or an enemy deals to his enemies each second.

Externalize: Hire part of the project (e.g.: the music) to another company.

Explosive damage: The maximum amount of damage a player or an enemy can deal in an instant

Feature: A characteristic of the game. E.g.: The player can jump, there is a level up system.

Feedback: Opinion from the game received from the testers.

Game Engine: A software that helps to create a video game.

Game genre: Classification of a video game depending on its features.

Game object: Anything in the game that the player can see or interact with. E.g.: a collectable, an enemy, a platformer, a background image, ...

Game pillars: The features that the game has and are taken into account in the game development process.

Hardware depreciation: The amount of money that every technological equipment loses every period of time. E.g.: a new computer does not cost the same when you buy it that a year after.

HUD (Heads Up Display): User interface when the player can see everything related to its character on the screen, such as its health bar, experience bar, skills buttons, ...

Industry standard: Criteria's from the game industry to know if the game is good or not.

Metroidvania Game: The name comes from the games Metroid and Castlevania. It is defined as a 2D platformer with an emphasis in exploring the world.

Milestone: A medium-long period of time at the end of which the indicated features of the game must be completed.

Newzoo: Website that contains videogame data.

NPC's (Non-Player Character): Characters in the game that are not players.

Parallax: The movement of the background images in order to create the sensation of movement. The furthest the background image is, the slower it moves.

Parent game object: A parent game object is a game object that has one or more child's game objects. Those child's inherit the transformation in world position from this parent.

Particle system: Technique used in games that allows to reproduce otherwise difficult phenomena's like fire by using a lot of small sprites.

Placeholders: A temporal replacement for something that will be changed in the future. E.g.: using a rectangle instead of a sword until the sword asset is completed.

Platformer Game: Games in which the player must jump between different platforms while avoiding enemies.

Playability: A measure that indicates if the game can be played or not. Not just in the sense of controlling the game, but also taking into consideration the quality of it.

Prototype: A first version of the game. The game will need a bit more of work to be finished for the marketplace. E.g. Some art assets are missing.

Published: Put the game in a store to be sold.

Q2: Second quarter of the year: April, May and June.

Replayability: Games that are worth playing more than once.

RPG (Role Playing Game): Game in which the player assumes the role of a character.

Scene: A file in which each level is stored.

Scene management: A way to manage transitions between scenes.

Script: A file that is executed by the game and controls some parts of it. E.g. A script that controls the player movement.

Soak idle testing: A test that consist in letting the game be playing for several hours to check if there is any problem if the game is used for an extended period of time.

Sound FX's: Sound Effects that are in the game.

Sprints (from Scrum): A relatively short period of time at the end of which the indicated tasks of the game must be completed.

Target / Target audience: People that the game is addressed to.

Tilemap sprite sheet: A set of sprites that are used to create a map.

True damage: Damage in a game that ignores the defense of the character.

UI (User Interface): Information from the game that appears on screen and the player uses to interact with it.

Unity: The game engine that I will use to develop this project.

Unity Asset Store: A store from Unity Engine where you can buy different assets. Some of them are free.

URL (Uniform Resource Locator): the address of a World Wide Web.

User friendly: Easy to use.

1. Introduction

1.1 Motivation

Since I started this degree, I always wanted to create an RPG game on my own, but I could not do that before for various reasons.

In the beginning of the degree, I had not enough knowledge to create it. It was not possible neither to redirect other projects I made in the degree in that direction, and they also take me most of my time, so I could not do it on my free time.

Because of those reasons it would be perfect for me to make this personal project in my TFG, as I wanted to do it since some years ago.

1.2 Problem formulation

Use the knowledge I learned in the bachelor's degree in order to develop and design a prototype of a videogame using the engine Unity.

The main goal of the game is to be fun to play. Other things like creating a good narrative are not as relevant in the project.

The art of the game will be bought as my objectives and motivations are not related with art.

1.3 General objectives

Creating a prototype of a 2D, RPG, action and platformer game using Unity with cartoon esthetics for computer.

1.4 Specific objectives

- Design from scratch the whole game.
- Develop a prototype of a videogame on my own.
- Learn to manage all my tasks to carry out the project satisfactorily.
- Achieve a fun gameplay for the players, known by the playtesting's that will be made.
- Learn to balance the game in order to make it fun to play.
- Implement an item system including its UI good enough for the industry standards.
- Carry out a useful playtesting and get enough feedback.

1.5 Scope of the project

The final scope of this project is to check if the prototype created is accepted for the main target and if it is well balanced with their feedback.

And who is the main target? The game will have a PEGI 12 rating: The game will have violence but will not be specific and it would be non-realistic violence as it is posted in the official PEGI 12 rating. Other analyzed games have a bit bigger PEGI rating as they include more explicit violence, but in this case, the main target is a bit younger, a male between 14 and 25 years.

Using this prototype, a final game can be created with a few months of work and even be published. Once published, every usual player from this genre can play and benefit from the project and even some players who are not our main target.

2. State of the art

Video games are growing year after year and it does not seem to stop, at least in some years. In 2018, the industry generated 134.900 million dollars, growing a 10.9%, according to Newzoo.

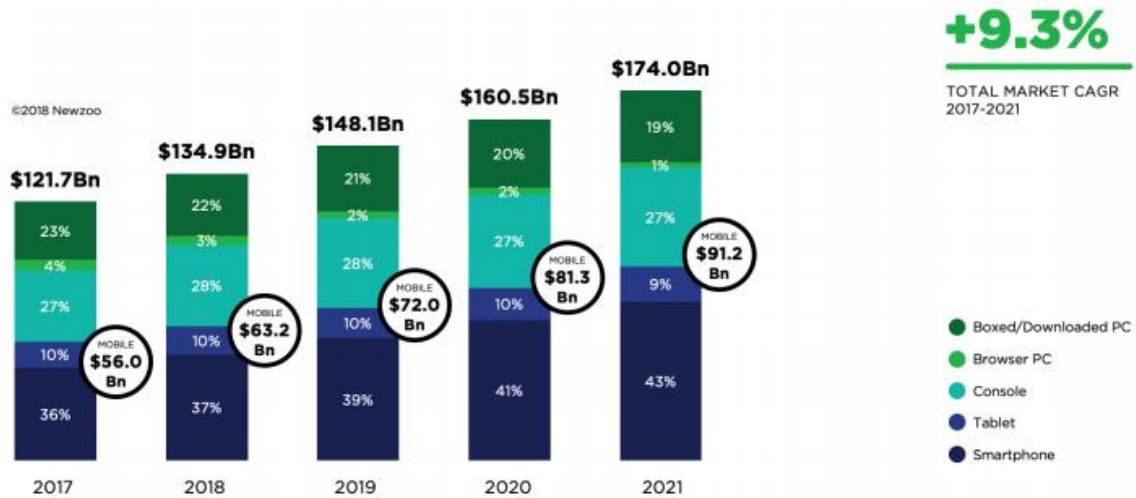


Figure 1: Global games market (font: White book DEV 2018)

As the money that moves the industry is huge, there are also a lot of companies. The biggest are:

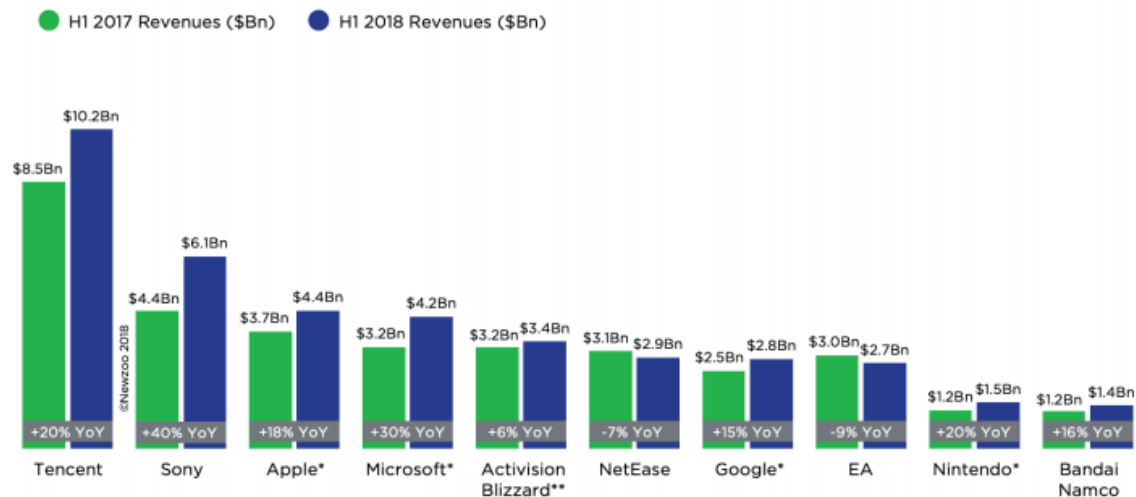


Figure 2: Top video game companies by revenue (font: White book DEV Spain 2018)

In the case of Spain, Catalonia is the place where most companies are, mainly in Barcelona, where there are Ubisoft, Gameloft, King, Zeptolab, Scopely, Riot Games, Tilting Point, IGG, Outfit7, among others.

It is followed by Madrid, where, except for Electronic Arts and PlayStation Spain, there are mainly local studies like Mercury Steam, Tequila Works, CremaGames, Fictorama, Gammera Nest, among others.

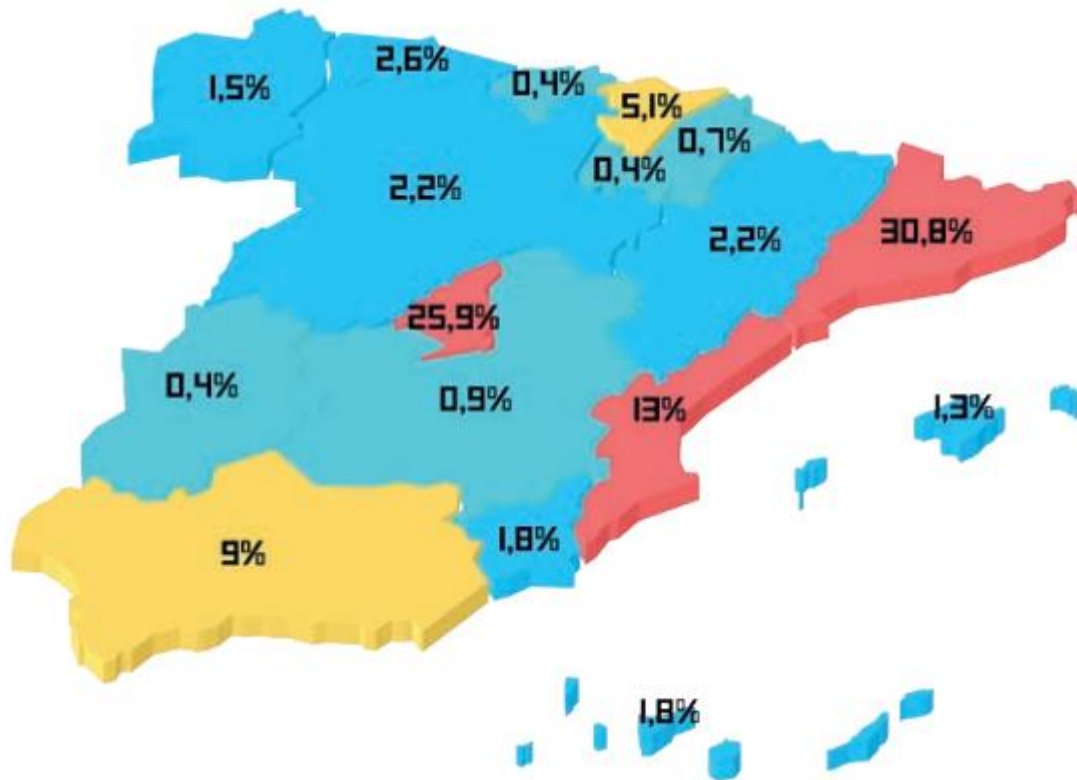


Figure 3: Territorial distribution of game companies in Spain (font: White book Spain DEV 2018)

The inversions in the industry are on the rise as well. In 2018 there has been a total of 5.000 million dollars in video games companies worldwide.

2.1 Technology

In Spain, Unity engine is the most used by far: 85% of the companies use Unity, followed by Unreal Engine with a 23%. Consider that some companies use more than one engine, that is why the percentages do not add up to 100%.

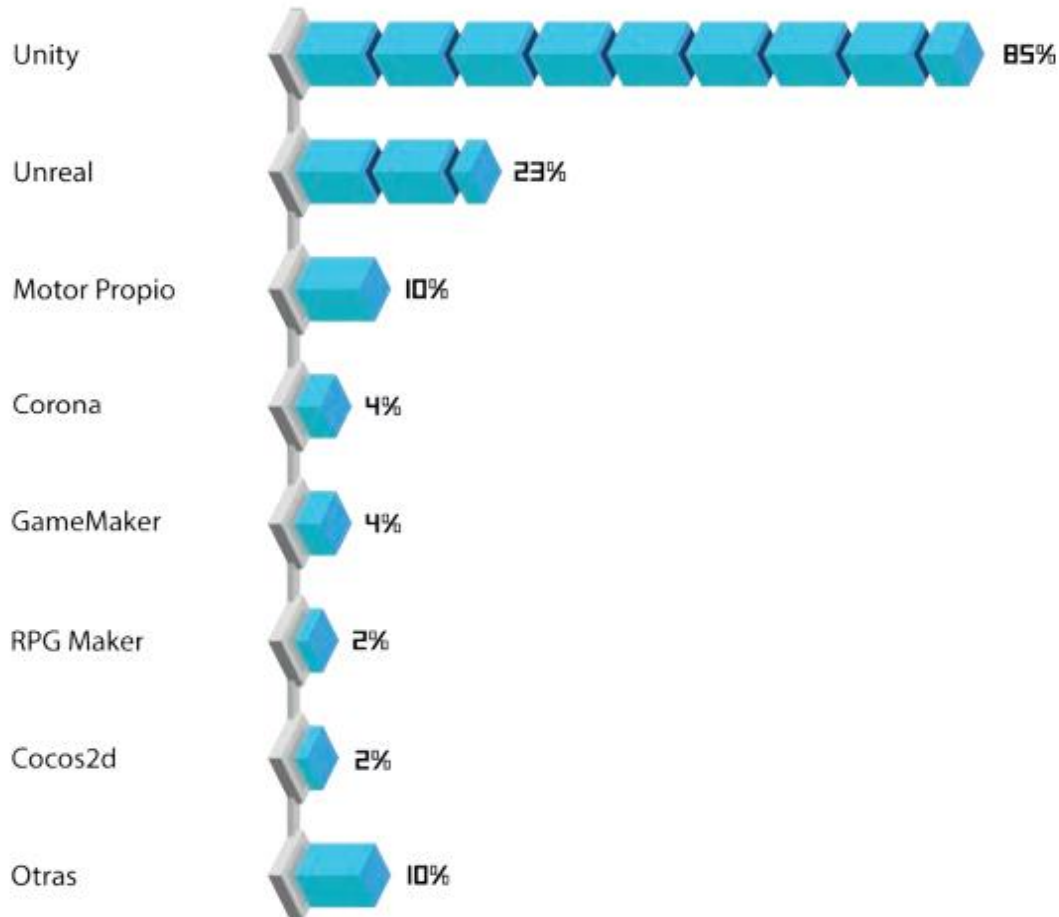


Figure 4: Software used to develop videogames by companies in Spain (font: White book Spain DEV 2018)

2.1.1 Unity

Unity released his 2018.3 beta version on September 14, 2018 and was released officially on December 13, 2018. This is the version that will be used in this project.

The Unity engine is an extensive software. It contains a huge number of functionalities in the development of 2D and 3D games:

- Includes an AI navigation tool so the characters of the game do not get stuck in places.
- Creating a UI is fast and efficient.
- Creating instances of everything that you want to in the game is easy, allowing to make fast changes to the levels.
- Includes an integrated physics engine
- Allows a fast iteration testing the game in the editor.

- Extendible, thanks to the community and the assets downloadable in the Unity Asset Store.
- Allows the creation of particle systems, giving the game much more visual interest.
- Includes animations control.
- Allows real time audio mixing.
- Optimized CPU and memory usage.

2.1.2 Unreal Engine

Unreal Engine released his 4.21 version on November 7, 2018.

Unreal Engine is also an extensive software, but less intuitive than Unity. It includes everything that Unity has but is more orientated to game development of bigger teams.

It provides better graphic capabilities than Unity, but it is slower to produce content, which makes it a bit worse specially at prototyping.

2.2 Market research study

An extensive market research study has been done to decide which software and programming language is going to be used, if the game can be successful, and to be conscious about how far the project can go.

The selected similar games to consider follow those criteria's:

- That they are the same or very similar genre
- That they are considered good games by the players
- That they are made for PCs

The games considered are:

- Catmaze
- Dead Cells
- Guacamelee! 2
- Salt and Sanctuary
- Timespinner
- Treasure Adventure World
- Unworthy
- Warlocks 2: God Slayers
- Wuppo

The first table is about basic information from those games:

Game	Genre	Release date	Development place
	Metroidvania, Platformer	24.05.2018	Saint Petersburg, Russia
	Metroidvania	06.08.2018	Bordeaux, France
	Metroidvania, platformer	21.08.2018	Toronto, Canada
	2D, Action, RPG	15.03.2016	Seattle, Washington, USA
	Metroidvania, RPG, Action	25.09.2018	Seattle, Washington, USA
	Metroidvania, Platformer	07.02.2018	Austin, Texas, USA
	Metroidvania, RPG, Action	29.05.2018	Toronto, Canada
	Metroidvania, Action	Q2 2019 (not released yet)	Poznan, Poland
	Platformer, Action	29.09.2016	Netherlands

Table 1: Similar games basic information

In order to make the following scores table of those games, the pages that has been considered are Steam, Metacritic and IGN. The reason to choose those platforms for the score and not others is because they are a reference for a lot of players.

In the case of Metacritic, there are two differentiated scores: the scores from their critics, and the scores from their users; both scores are differentiated in their website and so are in this table.

Metacritic scores are between 0 and 100, but in the table, I homogenized them to be between 0 and 10 to match the other reviews:










Game	Steam	Metacritic	Metacritic users	IGN	Average
	9	-	-	-	9
	9	8.7	7.5	-	8.4
	9	8.3	8	9	8.58
	9	8.4	8.1	8.2	8.43
	9	8.1	5.4	-	7.5
	9	-	-	-	9
	9	-	-	-	9
	-	-	-	-	-
	10	7	6.2	-	7.73

Table 2: Similar games score

A development analysis of those games has been made as well.

In the case of Timespinner, there were actually one developer, but the narrative and the sounds were externalized, that's why in the table appear as 3 developers.

On the other hand, the game Warlocks II: God Slayers is predicted to be finished in 2 years, but as delays are common in the industry, it may be longer.

The Catmaze case is also a bit particular, as seen in the table, the development time is so low compared to the others, especially because it was just one developer alone. That is because the engine he used (Construct 2) is a really simplistic way to develop, where you do not really program anything, the engine gives all the code for you, making it really faster to produce a game, but having much less control over it.

The same thing happens with Wuppo, which used Game maker and it is more or less the same, however, these developers were working on regular jobs while developing Wuppo, that is the reason why it seems a normal development time like the rest of the games.




Game	Engine used	Programming language	Number of developers	Development time
	Construct 2	Javascript	1	≈ 1 year
	Heaps	Haxe	9	3 to 4 years
	Own engine	-	12	2 years and 2 months
	Own engine using Monogame	C#, HLSL	2	≈ 3 years
	Own engine	-	3	4 years and 3 months
	-	-	5	≈ 6 years
	Own engine	-	4	≈ 2 years and 8 months
	Unity	C#	5	≈ 2 years
	Game maker	GML	3	≈ 2 years

Table 3: Similar games development information

As we can see, these similar games have not used a specific engine or programming language to develop their game.

Most of them made their own engine, however, this option is non-viable because it would be too much time to complete the project.

As there does not exist a standard better option, I decided to create my game using Unity, as it is an engine I used 3 times, and this allows me to create a better prototype and a decent game that otherwise, with another engine will not be possible. This includes using C# as it is the default in Unity and allows me to create everything I want to.

3. Project management

3.1 Procedures and tools for the tracking of the project

3.1.1 GANTT

	Starting week	Ending week	Weeks, monday to monday																			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
			11 February	18 February	25 February	4 March	11 March	18 March	25 March	1 April	8 April	15 April	22 April	29 April	6 May	13 May	20 May	27 May	3 June	10 June	17 June	24 June
Deliveries date	1	20																				
Rubric 1	1	5																				
Rubric 2	6	12																				
Rubric 3	13	17																				
Bachelor's thesis delivery	18	20																				
Memory	1	19																				
Introduction	3	3																				
State of art	1	2																				
Project management	1	2																				
Methodology	3	4																				
Project development	5	17																				
Conclusions	18	19																				
Research	1	5																				
Bachelor's thesis lessons	1	4																				
Competence research	1	3																				
Art assets research	3	5																				
Design	5	16																				
Player	5	5																				
Enemies	5	5																				
Maps	5	5																				
UI (User Interface)	5	5																				
Items	5	5																				
1st map tileset	5	5																				
1st map inside Unity	6	6																				
2nd map tileset	6	6																				
2nd map inside Unity	16	16																				
3rd map tileset	6	6																				
3rd map inside Unity	16	16																				
Environment sounds/FX's	5	5																				

Table 4: GANTT 1

	Starting week	Ending week	Weeks, monday to monday																			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
			11 February	18 February	25 February	4 March	11 March	18 March	25 March	1 April	8 April	15 April	22 April	29 April	6 May	13 May	20 May	27 May	3 June	10 June	17 June	24 June
Programming	6	20																				
Players animations	6	6																				
Enemies animations	7	7																				
Player base movement	6	6																				
Put enemies in game	7	7																				
Enemies AI	9	11																				
Player skills	6	8																				
Enemies skills	10	11																				
Level up system	8	8																				
Items system	12	12																				
Release playtesting	13	13																				
Release Delivery 3	17	17																				
Polish	18	19																				
Final Release	20	20																				
Main menu UI	13	13																				
Player level up UI	8	8																				
Skills UI	8	8																				
Items UI	12	12																				
Environment sounds	6	6																				
Player FX's	9	9																				
Enemies FX's	11	11																				
UI FX's	13	13																				
World FX's	6	6																				
Testing																						
Enemies AI	12	12																				
Combat	12	13																				
Google form for playtesting	13	13																				
Select playtesting users	13	13																				
Personal playtesting	12	13																				
External playtesting	14	14																				
Analyze external playtesting	15	15																				
Bugs	7	17																				
Combat	12	14																				
Post-Testing bugs	15	16																				
Other probable bugs	7/15	11/17																				

Table 5: GANTT 2

3.1.2 Personal timetable

This personal timetable allows to check that I have enough time to complete the project with the estimated hours dedicated to each task, to ensure the success of the prototype.

Weeks until final delivery	20	Real hours/week	20
Minimum hours/week	15	Total hours until final delivery	400

INTERNSHIP

LUNCH/DINNER

WORKING

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
08:00 - 08:30							
08:30 - 09:00							
09:00 - 09:30							
09:30 - 10:00							
10:00 - 10:30							
10:30 - 11:00							
11:00 - 11:30							
11:30 - 12:00							
12:00 - 12:30							
12:30 - 13:00							
13:00 - 13:30							
13:30 - 14:00							
14:00 - 14:30							
14:30 - 15:00							
15:00 - 15:30							
15:30 - 16:00							
16:00 - 16:30							
16:30 - 17:00							
17:00 - 17:30							
17:30 - 18:00							
18:00 - 18:30							
18:30 - 19:00							
19:00 - 19:30							
19:30 - 20:00							
20:00 - 20:30							
20:30 - 21:00							
21:00 - 21:30							
21:30 - 22:00							

Table 4: Personal timetable

3.1.3 Hack N Plan

A good management is a must in the project, for this reason, the software *Hack N Plan* will be used to have all the tasks controlled in the following lists: to be done, in progress, on testing and completed.

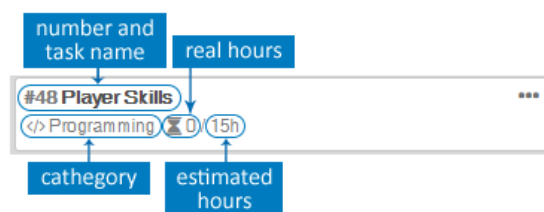


Figure 5: Hack N Plan task example (self-made)

This software had been chosen because it is specially designed for video games and has everything needed to have a good management and even some graphs with metrics of the progress, to see if the timing is going well or not.

Other management tools like *Trello* were discarded because they do not have all the things that I need in order to save time like time management.

While others like *Monday* are not the best option neither. Despite they have everything I need as *Hack N Plan*, they are not oriented to the video game industry and they are not free.

3.1.4 Bitbucket repository

In order to control versions and have everything on the internet, a Bitbucket repository will be used.

As I bought some assets, I cannot make a public repository; and although personally I am more used with Github, there is no option to have a private repository without paying.

That's the reason to use Bitbucket instead.

3.2 Evaluation tools

3.2.1 Personal and close people testing's

In order to find especially possible bugs and, in some degree, if the game is funny or not and how to improve it.

3.2.2 Playtesting between target audience

As soon as a playable build exists, it will be sent to a group of players together with a Google Form in order to get feedback of the game, the things that are going well, and most importantly, the things that have to be improved.

The main objective for this playtesting's is to see if the game is fun to play and how to make it better, whether it is fun already or not.

However, those playtesting sessions probably will make some bugs appear as well.

3.3 SWOT

	Positives	Negatives
Internal origin	<p>Strengths</p> <p>As I work alone, it is easier to manage the project and the work does not depend on anyone.</p> <p>There are many other games that combine these genres that are successful.</p> <p>Buying all art assets makes this prototype look way more beautiful and can save a lot of time.</p>	<p>Weaknesses</p> <p>I am the only one working on the game, so it costs more time to get everything done.</p> <p>The game might not be developed optimally because I have only developed 3 games in Unity.</p> <p>Once the prototype is finished, if I want to develop the full game I would most likely need an artist.</p>
External origin	<p>Opportunities</p> <p>The cartoon esthetic is something new in a 2D RPG, platformer and action video game, this can attract new players.</p> <p>There are few big companies creating this kind of games.</p> <p>BGW (Barcelona Games World) dates are near the final game release if I continue developing the game until the end to show the new game to probable players.</p>	<p>Threats</p> <p>The final game can be done faster by a company if they get the main idea of this prototype.</p> <p>There are other games that, despite not having the same esthetics, have a similar playability.</p>

Table 6: SWOT

3.4 Risks and contingency plan

Those are the possible risks considered and its solutions from lower to major importance:

Risk	Solution
Not enough art assets.	As it is a prototype, placeholders will replace the final art.
I may have problems with the Unity version I am working with.	Use an older stable version of Unity.
More work than expected: Not enough time to complete everything.	Some of the tasks will be deleted. E.g.: the prototype can be done in just one map.
Game is not funny enough.	With the planned playtesting this can be solved. This can cause more work and even the risk mentioned above.
My computer breaks.	In that case I will need to buy another one as I need to have one anyway.

Table 7: Possible risks

3.5 Initial costs analysis

The costs analysis consisted in an approximation where 3 different cases were considered: The best case, the probable case and the worst case.

All the graphics and tables (except *Table 11: Similar games market price*) are from a Google Sheet. In order to view it you can click on any image from this part.

The planning has been made considering that the prototype is finished in June and the game is finished in August.

Here we have some graphs with the net cumulative result in a 1-year time view since February 2019 to February 2020, which starts by the development of this prototype, and ends after 6 months of the finished game is released, passing by the end of the prototype (end of this bachelor's degree), and the end and publish of the game for the future work.

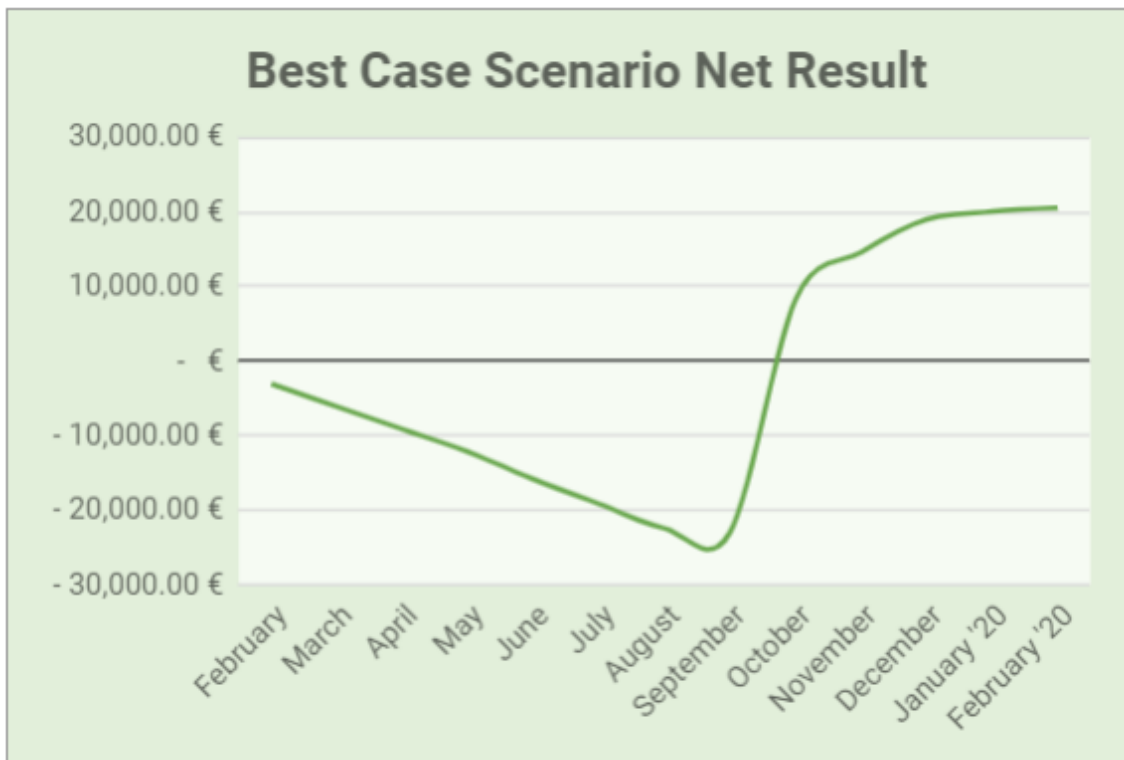


Figure 6: Best case scenario net results (self-made)

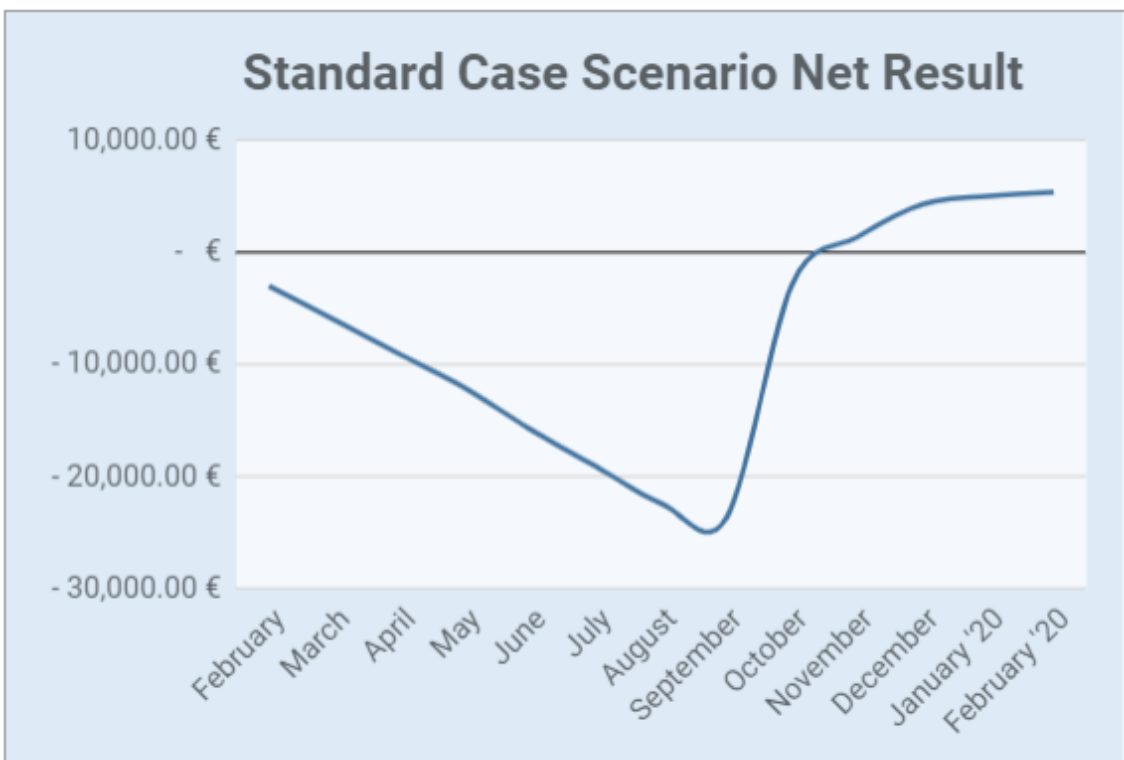


Figure 7: Standard case scenario net result (self-made)

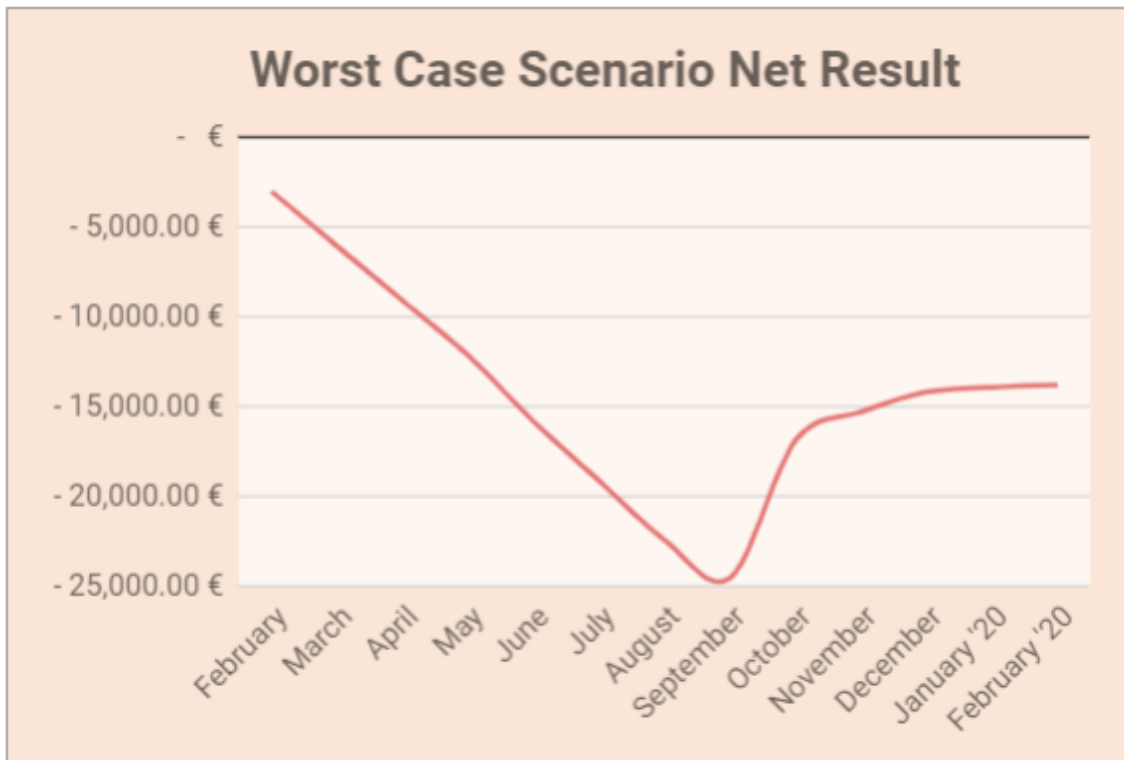


Figure 8: Worst case scenario net result (self-made)

Those graphs come from the following tables:

	February '19	March '19	April '19	May '19	June '19	July '19	August '19	September '19	October '19	November '19	December '19	January '20	February '20
TOTAL INCOME	- €	- €	- €	- €	- €	- €	- €	2,098.95 €	31,484.25 €	6,296.85 €	4,403.70 €	1,049.48 €	524.74 €
TOTAL EXPENSES	3,054.55 €	3,033.55 €	3,033.55 €	3,033.55 €	3,683.55 €	3,293.55 €	3,293.55 €	2,643.55 €	200.00 €	- €	- €	- €	- €
NET RESULT	- 3,054.55 €	- 3,033.55 €	- 3,033.55 €	- 3,033.55 €	- 3,683.55 €	- 3,293.55 €	- 3,293.55 €	- 544.60 €	31,284.25 €	6,296.85 €	4,403.70 €	1,049.48 €	524.74 €
CUMULATIVE NET RESULT	- 3,054.55 €	- 6,088.10 €	- 9,121.65 €	- 12,155.20 €	- 15,838.75 €	- 19,132.30 €	- 22,425.85 €	- 22,970.45 €	8,313.80 €	14,610.65 €	19,014.35 €	20,063.83 €	20,588.56 €

Table 8: Net profit best case scenario

	February '19	March '19	April '19	May '19	June '19	July '19	August '19	September '19	October '19	November '19	December '19	January '20	February '20
TOTAL INCOME	- €	- €	- €	- €	- €	- €	- €	1,399.30 €	20,989.50 €	4,197.90 €	2,935.80 €	699.65 €	349.83 €
TOTAL EXPENSES	3,054.55 €	3,033.55 €	3,033.55 €	3,033.55 €	3,683.55 €	3,293.55 €	3,293.55 €	2,643.55 €	200.00 €	- €	- €	- €	- €
NET RESULT	- 3,054.55 €	- 3,033.55 €	- 3,033.55 €	- 3,033.55 €	- 3,683.55 €	- 3,293.55 €	- 3,293.55 €	- 1,244.25 €	20,789.50 €	4,197.90 €	2,935.80 €	699.65 €	349.83 €
CUMULATIVE NET RESULT	- 3,054.55 €	- 6,088.10 €	- 9,121.65 €	-12,155.20 €	-15,838.75 €	- 19,132.30 €	- 22,425.85 €	- 23,670.10 €	- 2,880.60 €	1,317.30 €	4,253.10 €	4,952.75 €	5,302.58 €

Table 9: Net profit standard case scenario

	February '19	March '19	April '19	May '19	June '19	July '19	August '19	September '19	October '19	November '19	December '19	January '20	February '20
TOTAL INCOME	- €	- €	- €	- €	- €	- €	- €	524.74 €	7,871.06 €	1,574.21 €	1,100.93 €	262.37 €	131.18 €
TOTAL EXPENSES	3,054.55 €	3,033.55 €	3,033.55 €	3,033.55 €	3,683.55 €	3,293.55 €	3,293.55 €	2,643.55 €	200.00 €	- €	- €	- €	- €
NET RESULT	- 3,054.55 €	- 3,033.55 €	- 3,033.55 €	- 3,033.55 €	- 3,683.55 €	- 3,293.55 €	- 3,293.55 €	- 2,118.81 €	7,671.06 €	1,574.21 €	1,100.93 €	262.37 €	131.18 €
CUMULATIVE NET RESULT	- 3,054.55 €	- 6,088.10 €	- 9,121.65 €	- 12,155.20 €	- 15,838.75 €	- 19,132.30 €	- 22,425.85 €	- 24,544.66 €	- 16,873.60 €	- 15,299.39 €	- 14,198.46 €	- 13,936.09 €	- 13,804.91 €

Table 10: Net profit worst case scenario

In order to calculate an average price for the sells, a little analysis on similar games has been made.

Game	Price (€)
	9.99
	24.99
	19.99
	17.99
	19.99
	7.99
	13.49
	Not released
	14.99

Table 11: Similar games market price

With this data here is the estimated income:

	February '19	March '19	April '19	May '19	June '19	July '19	August '19	September '19	October '19	November '19	December '19	January '20	February '20
Number Sales	0	0	0	0	0	0	0	100	1500	300	600	50	25
Earnings Before Taxes	- €	- €	- €	- €	- €	- €	- €	1,999.00 €	29,985.00 €	5,997.00 €	4,194.00 €	999.50 €	499.75 €
Distribution Fee	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%
TOTAL INCOME	- €	- €	- €	- €	- €	- €	- €	1,399.30 €	20,989.50 €	4,197.90 €	2,935.80 €	699.65 €	349.83 €

Game Price	19.99 €
Christmas Sale Price	6.99 €

	Unit Sales	Pricing
BEST CASE SCENARIO	1.5	1
STANDARD CASE SCENARIO	1	1
WORST CASE SCENARIO	0.5	0.75

Table 12: Estimated income

And the estimated costs:

		February '19	March '19	April '19	May '19	June '19	July '19	August '19	September '19	October '19	November '19	December '19	January '20	February '20
HHRR	Total (With IRPF)	2,990.00 €	2,990.00 €	2,990.00 €	2,990.00 €	3,640.00 €	3,250.00 €	3,250.00 €	2,600.00 €	- €	- €	- €	- €	- €
	Programmer	1,000.00 €	1,000.00 €	1,000.00 €	1,000.00 €	1,000.00 €	1,000.00 €	1,000.00 €	1,000.00 €	- €	- €	- €	- €	- €
	Designer	1,000.00 €	1,000.00 €	1,000.00 €	1,000.00 €	1,000.00 €	1,000.00 €	1,000.00 €	1,000.00 €	- €	- €	- €	- €	- €
	Musician	300.00 €	300.00 €	300.00 €	300.00 €	300.00 €	- €	- €	- €	- €	- €	- €	- €	- €
	Artist	- €	- €	- €	- €	- €	500.00 €	500.00 €	500.00 €	- €	- €	- €	- €	- €
Suppliers	Art bought	21.00 €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €
Office	Rental	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €
	Office Material	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €
	Electricity	19.80 €	19.80 €	19.80 €	19.80 €	19.80 €	19.80 €	19.80 €	19.80 €	- €	- €	- €	- €	- €
	Hardware Depreciation	23.75 €	23.75 €	23.75 €	23.75 €	23.75 €	23.75 €	23.75 €	23.75 €	- €	- €	- €	- €	- €
Production Expenses	Unity	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €
	Google Drive	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €
	HacknPlan	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €
Marketing	Barcelona Games World Exposition	- €	- €	- €	- €	- €	- €	- €	- €	200.00 €	- €	- €	- €	- €
TOTAL COSTS		3,054.55 €	3,033.55 €	3,033.55 €	3,033.55 €	3,683.55 €	3,293.55 €	3,293.55 €	2,643.55 €	200.00 €	- €	- €	- €	- €

Table 13: Estimated costs

In the part of costs, the hardware depreciation is calculated by the following:

	Initial date	Final Date	Asset value	Amortization period (years)	Months	Amortiz./anual	Monthly Depreciation	Special Conditions
Computers	2/14/2019	2/14/2023	1,000.00 €	4	48	250.00 €	20.83 €	-
Screens	2/14/2019	2/14/2024	100.00 €	5	60	20.00 €	1.67 €	-
Keyboards	2/14/2019	2/14/2024	75.00 €	5	60	15.00 €	1.25 €	-
TOTAL			1,175.00 €			285.00 €	23.75 €	

Table 14: Hardware depreciation

In this case, a lot of expenses are free because I am a student that works from home. Those costs would be 110€ each month for the Unity license, and the following for the office each month:

Rental	300.00 €	Electricity price		
Office Material	50.00 €	KW/h	€/KW	h/mes
Electricity	59.40 €	0.55	0.15	240
				Workers
				3

Table 15: Estimated office costs

3.6 Initial planification modification (05/04/2019)

There was a little bit of modification from the initial planification because after designing the game, there appeared some tasks that were not calculated on the start. Fortunately, as the methodology used demands testing and modifying things while developing, it was calculated an extra time for expected changes or more features needed, but they were a bit more expensive than it was though, so a little planning modification was needed.

Those problems consisted on the following:

3.6.1 HP bars on the enemies

Hp bars on the enemies were not planned, but they are necessary in order to give feedback to the player on how much health they have and how hard is to beat them.

To give even more feedback, for the last milestone, every character will have the number of damage received on screen for a short period of time every time they are hurt, giving the player even more feedback not only of the health and damage they have, but also if the hits are successful or not.

This caused a delay of 2 hours; and making the second part is calculated to have a work of 3 hours more.

3.6.2 Missing map sprites

New sprites added to the snow map were needed in order to have a realistic map and were also not expected. This caused a delay of 2 hours.

3.6.3 Missing particle effects

The player animations were really limiting, this was expected to be fine for a prototype but when it was the time to add skills to the player, the feedback from the animations became crucial. Creating the animations were way much time and not the objective of this project. For this reason, they were replaced by some particle effects. Creating them caused a delay of 4 hours.

3.6.4 Player respawning system

The initial idea could have been that the player has to win the level with only one life. This idea was cancelled as it seems too hard when the level was being created and then, a respawning system had to be created. This caused a delay of 2 hours.

3.6.5 Checkpoints

Along with the respawning system, the idea of checkpoints became obvious in some points of the levels and in the start of each one, so the player does not have to repeat the entire level if he dies. If not, the levels and boss fights could be repetitive. This caused a delay of 2 hours and it also caused some bugs with the player respawn that were solved afterwards.

3.6.6 Scene management

The major problem that was not calculated at the start was this one. Scene management was vital because all levels had to be loaded in different scenes. Loading everything in the same scene could have been possible at first, as it could be a little prototype, but to be sure people with bad computers could play the game with any problems, it was not possible anymore. The main problem of having everything in one scene is that the computer makes all the calculus at the same time and it can low the frames per second of the game.

Creating the levels on different scenes was not a problem at all, but saving the player stats, together with the HUD and other things, caused a lot of problems to be solved and caused a delay of 5 hours.

3.6.7 Maps interactions

The final think not planned that is still in progress is the map elements that interact with the player. Those consist on the following:

- Jumping platform
- Spikes
- Ice lance (finished)
- Coins (finished)
- Health potions (finished)

All of them will be explained in the design of the maps.

This caused a delay of 3 hours and a half, but it is calculated to be a couple of hours more to finish the spikes and the jumping platforms.

3.6.8 Assets remodeling

There were two assets that had been bought that had to be changed. This was not an artistic issue, if it was, then it will not have been changed; it was a design issue.

The first one was the hedgehog enemy. The change in this case was needed so he does not differ from the map and, as he does not seem dangerous, make him more similar to the map and some enemy. Making this change it is expected to be seen as an enemy for the players.

The other one was the Fluffy monster, which needed a bit more of remodeling, as it just rolls instead of walking and has a dynamite wick attached (because he explodes).

Both enemies will be explained in the Design section of the Development.

Those changes caused a delay of 3 hours and a half.

3.6.9 Balancing

Finally, there will also have to be a balancing, which will be in the end of the project. This is expected to take a lot of time because do a good balancing that feels right to the player and make the game not too easy neither too hard can be challenging. It is expected to take about 10 hours.

3.6.10 New GANTT

	Starting week	Ending week	Weeks, monday to monday																			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
			11 February	18 February	25 February	4 March	11 March	18 March	25 March	1 April	8 April	15 April	22 April	29 April	6 May	13 May	20 May	27 May	3 June	10 June	17 June	24 June
Deliveries date	1	20																				
Rubric 1	1	5																				
Rubric 2	6	12																				
Rubric 3	13	17																				
Bachelor's thesis delivery	18	20																				
Memory	1	19																				
Introduction	3	3																				
State of art	1	2																				
Project management	1	2																				
Methodology	3	4																				
Project development	5	17																				
Conclusions	18	19																				
Research	1	5																				
Bachelor's thesis lessons	1	4																				
Competence research	1	3																				
Art assets research	3	5																				
Particle effects research	8	8																				
Design	5	19																				
Player	5	5																				
Enemies	5	5																				
Maps	5	5																				
UI (User Interface)	5	5																				
Items	5	5																				
1st map tileset	5	5																				
1st map inside Unity	6	6																				
2nd map tileset	6	6																				
2nd map inside Unity	16	16																				
3rd map tileset	6	6																				
3rd map inside Unity	16	16																				
Environment sounds/FX's	5	5																				
Missing map sprites	6	6																				
Particle effects creation	8	8																				
Balancing	18	19																				

Table 16: New GANTT 1

	Starting week	Ending week	Weeks, monday to monday																			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
			11 February	18 February	25 February	4 March	11 March	18 March	25 March	1 April	8 April	15 April	22 April	29 April	6 May	13 May	20 May	27 May	3 June	10 June	17 June	24 June
Programming	6	20																				
Players animations	6	6																				
Enemies animations	7	7																				
Player base movement	6	6																				
Put enemies in game	7	7																				
Enemies AI	9	11																				
Player skills	6	8																				
Enemies skills	10	11																				
Level up system	8	8																				
Items system	12	12																				
Release playtesting	13	13																				
Release Delivery 3	17	17																				
Polish	18	19																				
Final Release	20	20																				
Main menu UI	13	13																				
Player level up UI	8	8																				
Skills UI	8	8																				
Items UI	13	13																				
Environment sounds	13	13																				
Player FX's	13	13																				
Enemies FX's	13	13																				
UI FX's	14	14																				
World FX's	13	13																				
Enemies HP bar's	7	7																				
Player respawn system	9	9																				
Checkpoints	9	9																				
Scene management	9	10																				
Map interactions	7/14	8/15																				
Testing	12	15																				
Enemies AI	12	12																				
Combat	12	13																				
Google form for playtesting	13	13																				
Select playtesting users	13	13																				
Personal playtesting	12	13																				
External playtesting	14	14																				
Analyze external playtesting	15	15																				
Bugs	7	17																				
Combat	12	14																				
Post-Testing bugs	15	17																				
Other probable bugs	7/15	11/17																				

Table 17: New GANTT 2

4. Methodology

An agile methodology is a must for this project, other methodologies are way too slow and they will almost assure a failure in this project because they do not have the fast iteration that is needed to make the game fun to play.

Scrum is the most popular agile methodology in software development and specially in video games, and this is because it helps to ensure that the goals are met with quality in each part of the development using the Sprints.

For those reasons the agile methodology Scrum will be used.

The following figure describes the process I will follow in a general term:

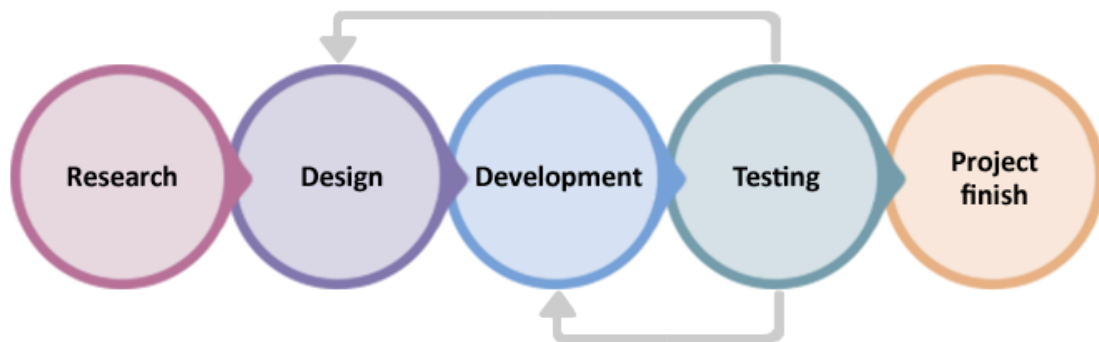


Figure 9: Methodology (self-made)

4.1 Research

The first things to be done will be the research. A good research is indispensable, and it is the first step for the project, specially information about competence and trying to deduce what makes them good games and how to translate this to our game.

After this, once the game idea is clear and it is decided what we will do, it is time to look for the art assets we want and plan all the management.

4.2 Design

Once everything is planned and the main goal is planned, the design must be done following the references chosen in the research. A GDD (Game Design Document) will not be made, everything will be made by one person, so there is no need to have everything in just one document, instead, there will be different documents explaining how each part of the game will be with a bit less detail than in a GDD.

The design includes the levels, the UI, the enemies, the player, the items, the music and the music effects.

4.3 Development

Right after the documents are finished and every detail is explained in them, the game can be started.

The first part in the development of the prototype will be creating the maps inside Unity. Once we have the first map we can start putting things on it, starting with the player, and including its skills and its level up system, together with the environmental sound and fx's.

Once the player can move inside the map and the level up system is made, the enemies will be added. Once we have both player and enemies, it is time to start building the interaction between them and developing the AI of the enemies, with their skills.

With this we can fight enemies, and with the combat system done we will most likely have bugs that must be solved as soon as possible. Once done, the main playability of the game will be finished.

4.4 Testing

This could be considered a really simplistic prototype and a simple playtest of the combat can be done; however, it is too early to look for people to play the game because it would be too time consuming in exchange of little feedback.

However, personal and close people playtesting must be done at this point. This way we can find not only bugs, but also start checking if the game is fun or not, and, in the case, it is not fun how to solve it. If it is the case, we have to go back to design and develop some features.

With this iteration we can improve the game faster and fail early in the project: when we have time to adapt and solve the problems.

While doing those testing's, all the final features of the prototype will be developed in parallel. This includes the item system, everything related to the UI (menu UI, skills UI, player leveling up UI and items UI) and the players and enemies fx's.

With the improvements from the iteration and with the final parts of the development are finished we can plan an external playtesting. At this point, the game would have past some iterations, so it is expected to be already good.

To plan this external playtesting and make it as useful as possible a Google Form will be made as a questionnaire to get good feedback from the testers.

Afterwards, with the result of the questionnaire, the prototype will be improved with these final bugs and improvements recollected by a larger number of testers and, finally, the prototype will be ready.

5. Project development

5.1 Assets research and selection

Before starting the project, a pack of assets was bought in the website www.humblebundle.com.

There was a huge amount of assets, and in order to organize all those assets and what to use and what not to, the first thing that was done is the selection of the assets that will be used in the game.

All those selected assets will appear in the following design section or in the game itself.

5.2 Design

Then, the design of everything had been done as planned.

5.2.1 Game design

The prototype consists in a 2D, RPG, action, platformer game with cartoon esthetics that has 3 different levels. Each level has his platforming phase with some enemies in between for the player to defeat or avoid, and in the end of those it has a phase

5.2.1.1 Game pillars

The game pillars taken into account every step on the design were the following:

- Gameplay
- Mobility
- Fast

5.2.1.1.1 Gameplay

The gameplay goes before anything, this means that if a decision in the design has to be made and it makes the gameplay worse, it will not be done or there will be another solution to the problem.

5.2.1.1.2 Mobility

Player mobility is a must, first of all because of the platformer part of the game, if he lacks mobility the design of the levels should be really poor and probably not fun.

5.2.1.1.3 Fast

The combat and the player movement has to be fast. This can affect specially the player and enemies design, making the combat more intense.

5.2.1.2 Game story

The story takes place in the Greek mythology. The player is an ambitious warrior of Sparta named Elyan. Sparta is currently in war and things are not going well for them and they are worried to lose everything.

However, one night he has a strange dream, Zeus itself appears in the dream and tells him about a way to win the war and not lose their beloved family. Zeus tells him that he will give him great powers if he can complete some quests for him.

Zeus explains to him that the quests consist in killing some of his enemies and that he can directly interfere with it and kill them because that would cause a war among Gods.

He has no doubt about accepting it because he does not want to lose everything he owns. To prove him that he can achieve those promised powers, Zeus gives him the power “Dark Shadows” that is in the game, explained in “5.2.3 Player design”.

The game name “Power chase” comes from this story about how your character tries to achieve new powers.

5.2.1.3 Flowchart

Those enemies from those quests are divided in different worlds and levels. In this playtest three levels have been created as planned and in the end of each level the player encounters the boss of the level, which is one of the quests from Zeus.

Each level is more difficult than the previous one, having its pikes at the boss battle.

The flowchart planned is the following:

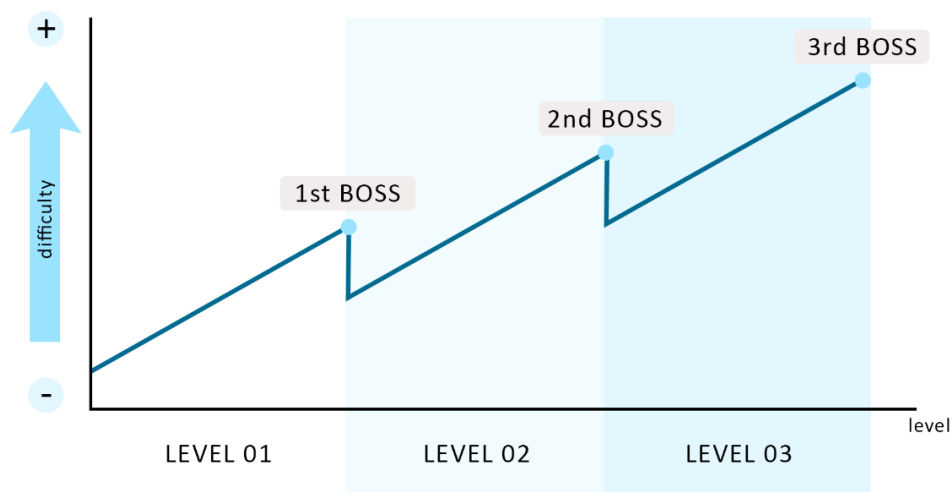


Figure 10: Flowchart (self-made)

5.2.2 Characters design

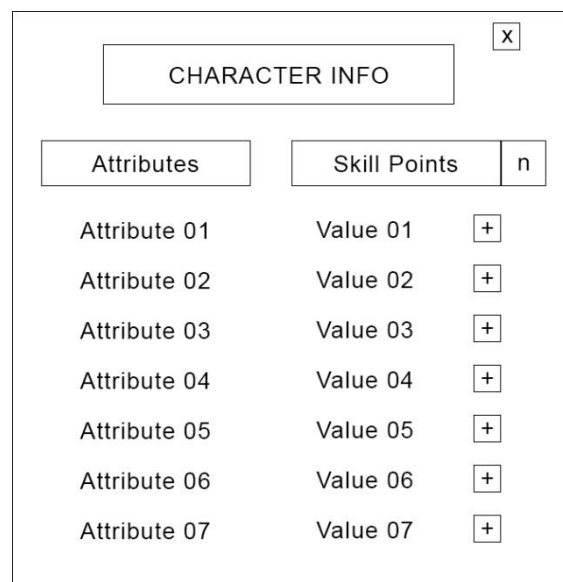
In order to design the combat, first the characters common attributes have to be defined. All of the characters have the same attributes:

- Movement speed: the speed at which the character moves in the map.
- Attack speed: the more speed the faster the attacks will be.
- Hit points (HP): The life of each character. If it falls to 0, the character dies.
- Attack: The damage the character deals to the others. It can be physic or magic damage, which determines which attribute the other character ignores.
- Critic chance: The probability an attack is multiplied by a factor.
- Critic multiplier: The factor by which a critic attack is augmented.
- Physic defense: The damage the character avoids if the attack was physic.
- Magic defense: The damage the character avoids if the attack was magic.
- Dodge: The probability that an attack is dodged.

However, there are some attributes only the player or the enemies have.

The enemies have the “given experience”, which is the experience that is given to the player when he kills them.

The player has its experience points and level. Experience points are obtained by slaying enemies, once certain amount of experience points are obtained, the player level increases by 1 and those points are reset. Each level gives the player one experience point that can be used in game to increase one attribute, which can be done using its interface:



Attributes	Skill Points	n
Attribute 01	Value 01	+
Attribute 02	Value 02	+
Attribute 03	Value 03	+
Attribute 04	Value 04	+
Attribute 05	Value 05	+
Attribute 06	Value 06	+
Attribute 07	Value 07	+

Figure 11: Character panel info design (self-made)

Each level needs more experience than the ones before following this equation:

$$Experience = BaseExperience * Factor^{(Level-1)}$$

- The base experience is the experience needed to level up the first time. 1000 in this case.
- The factor is a multiplier that in our case is 1.07, this factor is used in other games, while others use some other factor like 1.15.
- The level is the actual level of the player.

This equation was extracted from an article about the [design and math of incremental games](#).

5.2.2.1 Combat attributes example

Here is an example in order to clear out the characters attributes.

Player attributes:

- Attack: 100
- Critic chance: 25%
- Critic Multiplier: 2

Enemy attributes:

- Hit points (HP): 500
- Physic defense: 50
- Magic defense: 25
- Dodge: 10%

In this case, if the player hit the enemy, he has a base attack of 100, but there is a 25% chance (critic chance) that this damage is multiplied by 2 (critic multiplier), so there is a 75% chance to apply 100 damage and 25% to apply 200 damage.

In any case, the enemy has some of this damage reduced. First of all, there is a 10% chance (dodge) that he evades the attack and he receive no damage at all. If this does not happen, depending on the type of attack (physic or magic) the damage will be reduced by 50 if it is physic (physic defense) or by 25 if it is magic (magic defense). After this reduced damage, the 500 enemy hit points, will be reduced by the output of this calculations, and if it drops down to 0, the enemy will die.

Once the enemy dies, the player will receive the given experience that enemy has, and if it is enough to level up, he will.

5.2.3 Player design

The first idea for the player was to be the dwarf character, but it was changed later for a knight, which is better because the player will not be identified as a dwarf and probably lose interest in the game just at the start.

The idea was to have a dynamic and fast control on the player. For this reason, the main idea was to give an acceleration movement, but it was changed later because it was seen it made the opposite effect. The player can also double jump to give the sensation of more control on the air.

With the idea of having a dynamic control, not only fast movements but also fast casting skills were designed so the combat is funnier. For these reasons, the player basic attack can be done while in movement or even jumping and hits the enemies that are in right in front of him.

Also, following this idea of fast and dynamic combat, those skills were designed:

- Dash: The player can move in any direction really fast for a short period of time.
- Dagger throw: Throws a dagger in any direction that collides with the first enemy and damages him. It applies a base damage and a scaling factor depending on the attack of the player.
- Dark shadows: No casting time. It casts a black magic smoke from the player that damages all the enemies in a certain radius. The damage is reduced the more far the enemy is. It applies magic damage.

There are some other skills that are buffs for the player and can help him get through some phases:

- Sharp sword: Increases the attack of the player by some time.
- Extra armor: Increases the defense of the player by some time.
- Blood sword: For some time, a 15% of the damage inflicted to the enemies is returned to the player, healing him.
- Deflection: For some time, a 25% of the damage that the player receive is returned to the enemy who caused it.

There are some other skills that were canceled due to different reasons:

- Magic sword: If active, the player regular attack deals magic damage, otherwise it deals physic damage.
 - In this case, the focus of the game was not to be strategically correct and know which enemy has more physic or magic defense, as it went not along with the game, it was removed.
- Sword rain: 20 swords appear from the sky falling into the ground. If an enemy gets hit, it gets damaged.
 - This skill could have been in the game but was replaced by the implemented skill "Dark shadows".
- Berserk: Player goes frenzy, obtaining a boost that increases his movement speed, damage and reduce the damage taken for 30 seconds.
 - This skill was removed because it breaks the game in the sense that the player is way better in comparison and it will have no sense to fight bosses without the skill, which will make the bosses more like regular enemies.

5.2.4 Enemies design

Following the player, the enemies had been designed. They are divided between bosses and regular enemies.

There are more designed enemies that there are in the game because they seemed good at this point, just in the design, but then in the development after testing them, they seem repetitive from others or not as fun as expected.

At some point of the development, after testing the combat, it was decided that the enemies, instead of colliding between themselves, can overlap, so the combat is faster. The other way resulted in a really boring combat where almost every time you only have to fight one enemy at once.

5.2.4.1 Enemy bosses

Enemy bosses are enemies that the player encounters after finishing a level. They are way more powerful than regular enemies and have unique skills that nobody more has.

5.2.4.1.1 Monk

The Monk has a distance attack, its main attack is throwing a ball that he controls into the player that damages him. He does not move around the map, however he can teleport to different points to avoid the player and maintaining his distance with him.



Figure 12: Monk (bough asset)

He is the first boss that appears after the first level.

5.2.4.1.2 Ice Gorilla

The Ice Gorilla is a tough enemy that can jump over the player, dealing area damage when falling on the ground. He can also summon little Yeti's that have a similar behavior like him but are easier to deal with.

His normal attacks push the player backwards, making it harder to hit him.



Figure 13: Ice Gorilla (bough asset)

He is the second boss and appears after the player completed the second level.

5.2.4.1.3 Demon

The Demon is the biggest creature in the game, and so is his basic attack. It is a bit slow but has a lot of range and the player must be fast in order to dodge it.

He can also summon Imps and One-Eyed Imps to help him on the fight, and every 8 summonses, he absorbs the health of them, making him recover part of his health.



Figure 14: Demon (bough asset)

He is the third and final boss of this prototype and appears after the player completed the third level.

5.2.4.2 Regular enemies

Its behavior is simpler than the bosses and appear more than once in the different maps.

5.2.4.2.1 Ice Hedgehog

Its behavior is quite simple, it just goes from a point A to a point B and then goes back forever.

Occasionally, it throws 6 spikes in different directions that hurt the player if they hit him.

If the player collides with him, he also gets hurt and gets shoot up in the air.

He is not tough, but it is hard to hit with a normal attack because you can't get too close to him and neither stay near without moving because of the spikes he throws.



Figure 15: Ice Hedgehog (self-made)

The hedgehog sprite color was changed to adapt to the colors of the map. The original bought asset was this one:



Figure 16: Hedgehog (bough asset)

5.2.4.2.2 Archer

The archer is an enemy that shoots arrows to the player, it does not move from her position as she is on strategic elevated points in the map. She deals quite big damage but also has a low defense.



Figure 17: Archer (bough asset)

5.2.4.2.3 Dwarf

The dwarf is the base enemy from the game. He just follows the player when he sees him and has a close attack with his axe, quite similar to the player normal attack.



Figure 18: Dwarf (bough asset)

5.2.4.2.4 Bird

The bird is flying in the map and throws ice cubes on the ground that if they hit the player, he freezes for 1 second.



Figure 19: Bird (bough asset)

This enemy was removed from the game as it slowed the fast movement that the design aim and also made the player be really easy to kill if he got freeze.

5.2.4.2.5 Lizard monster

The lizard monster shoots his horn into the player and has a close attack when the player gets close to him.



Figure 20: Lizard monster (bough asset)

5.2.4.2.6 Fluffy monster

The Fluffy monster rolls over the map against the player and if he gets too close or after some seconds, it explodes, dealing damage in a range of 4 meters.



Figure 21: Fluffy monster (self-made)

The Fluffy monster design was also changed in order to make the player understand that he can roll over the map and that he can explode (because the dynamite wick). The original bought asset was this one:



Figure 22: Fluffy monster (bought asset)

5.2.4.2.7 Imp

The Imp is a simple enemy that follows the player in order to hit him in a close range. He is summoned by the Demon boss.



Figure 23: Imp (bough asset)

5.2.4.2.8 One-Eyed Imp

The One-Eyed Imp is a bit more difficult to beat than the regular Imp. He has a ranged attack where he shoots his horn into the player. He can be summoned by the Demon boss as well, and after some time he starts chasing the player like the regular Imp.



Figure 24: One Eyed Imp (bough asset)

5.2.4.2.9 Little Yeti

The Little Yeti has a jump attack that deals damage in a little area when he falls into the ground.



Figure 25: Little Yeti (bough asset)

5.2.5 Items design

In the case of items, there are two kinds, items that can be collected on the map, and items that can be bought on the Store.

5.2.5.1 Collectable items

Items that can be collected on the map are the following:

- Health potion: Can be accumulated by the player. Once he uses them, the potion immediately restores part of the player health.
- Coins: Coins can be collected and then used in the Store to buy the items in there.
- Key: The key is an item that the player needs to open the door that is in every level. This part is more detailed in the next section, Levels design.

5.2.5.2 Items on the Store

Items that are on the store can be bought by the player in exchange of an amount of coins. Those items upgrade the player attributes when bought and are divided by different equipment.

- Weapons: Increase the player attack damage, and some of them the critic chance and the critic damage.
- Helmets, armors and belts: Increase the player HP and the physic or magic defense depending on the item.
- Gauntlets: Increase the HP, the physical defense and the attack of the player.
- Boots: Increase the movement speed of the player and can also increase the physical defense.
- Necklace: Increase the dodge of the player.

Health potions can also be upgraded on the Store. Each time they are upgraded they restore more health to the player when they are used.

5.2.6 Levels design

Every level follows some steps that the player needs to follow to advance. The end of the level is a door that leads to a Boss fight, but in order to open the door, the player needs to find the Key first, which is an item that will be in some point of the map.

Making this means that the levels will follow this cycle:

Start level → Find the Key → Open the door → Fight a Boss → Start new level and repeat

Levels have some interactable objects:

- Coins and health potions that the player can collect.
- Moving platforms, pretty self-descriptive, platforms on the level that move following a path.

- Ice lances, which are places in the map that shoot ice lances in a direction that if they hit the player, they apply damage to him.
- Spikes, which are a part of the map that if the player touches, he gets hurt.
- Jumping platforms, a place in the map that if the player touches he gets launched into the air to reach for high places.

The levels were made at the first place on paper, as it is easier, faster, and it is easier to have new ideas drawing by hand rather than on a computer. It is important to remember that this was one of the first iterations, which means that the final maps can be a bit different, although they follow the same idea.

The level goes from left to right and the end of a part is the beginning of the following part.

5.2.6.1 Level 1



Figure 26: Level 1 Part 1 (self-made)

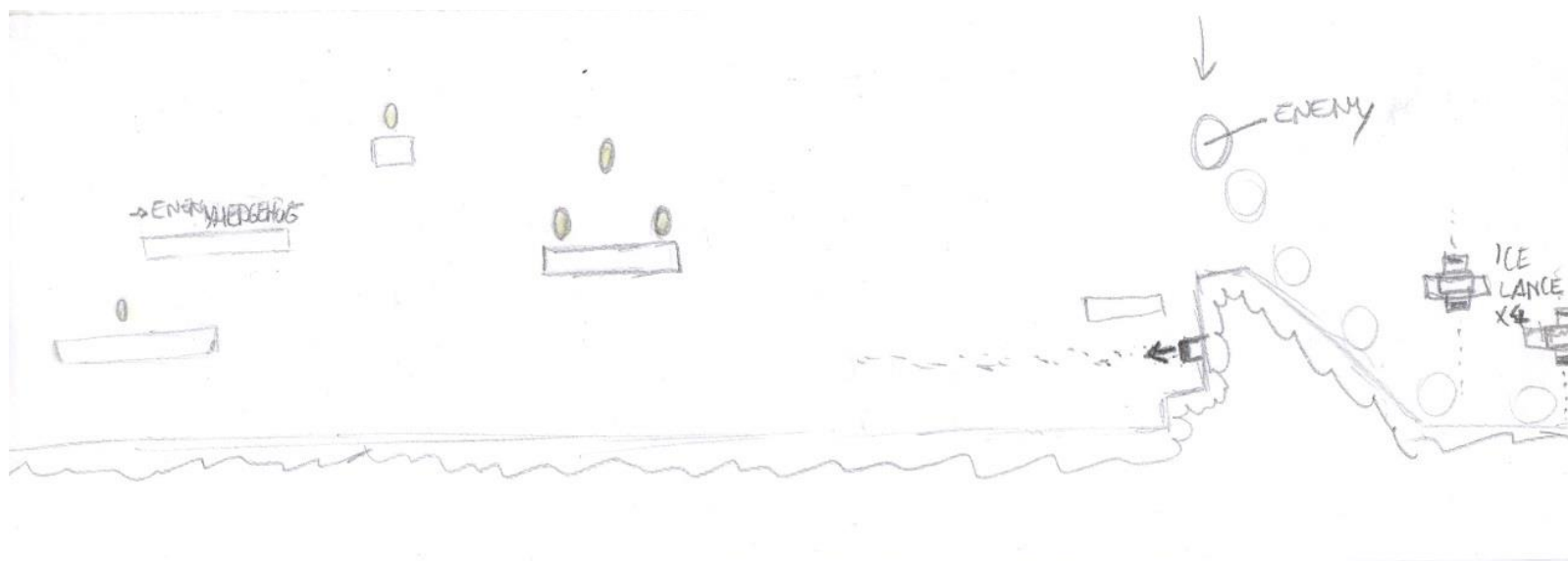


Figure 27: Level 1 Part 2 (self-made)



Figure 28: Level 1 Part 3 (self-made)

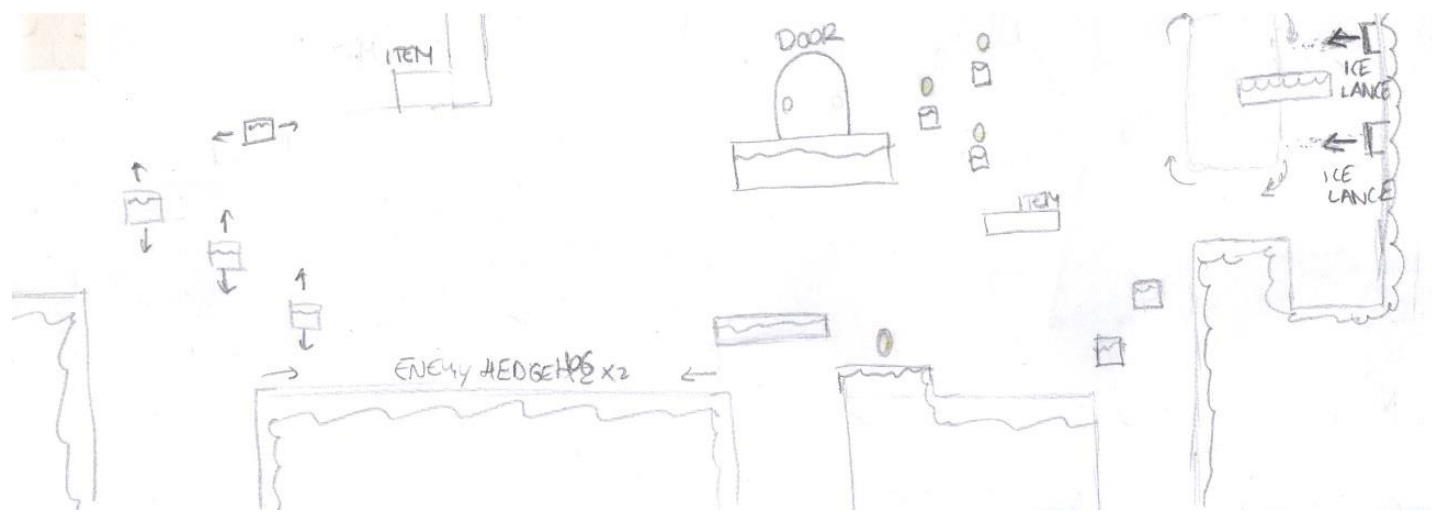


Figure 29: Level 1 Part 4 (self-made)

5.2.6.2 Boss Level 1

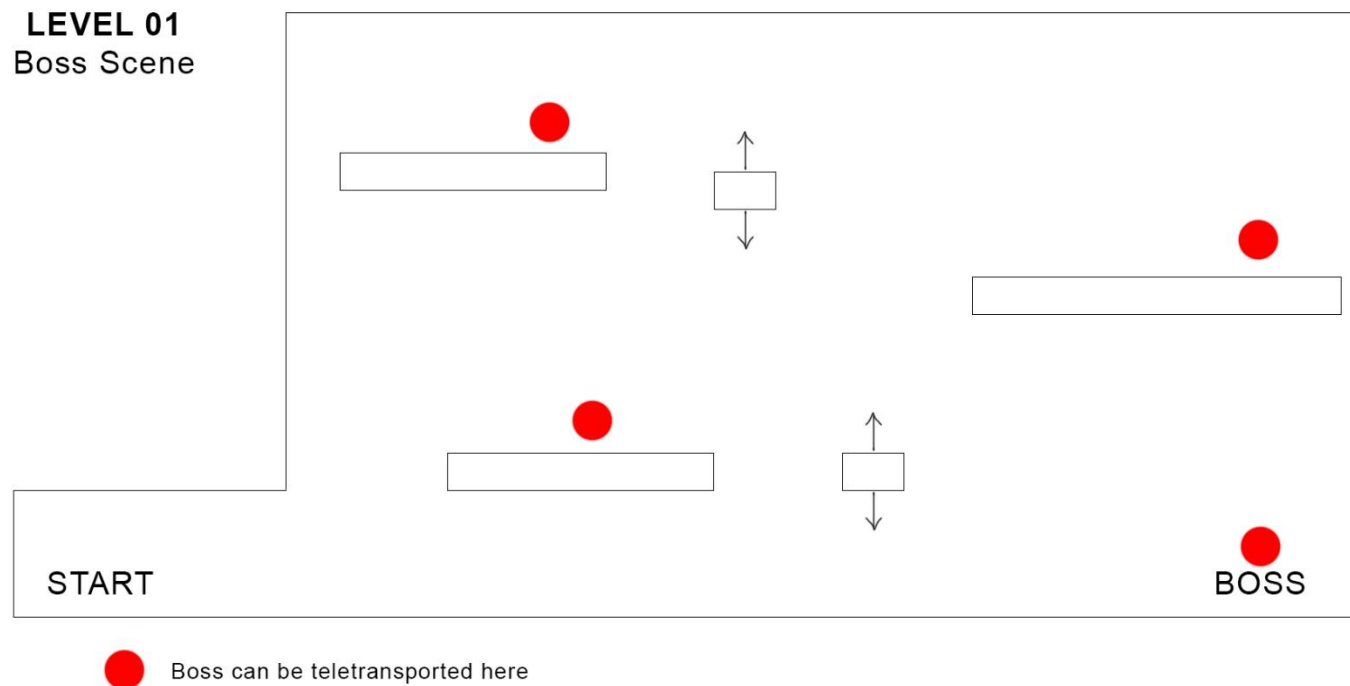


Figure 30: Boss level 1 (self-made):

5.2.6.3 Level 2

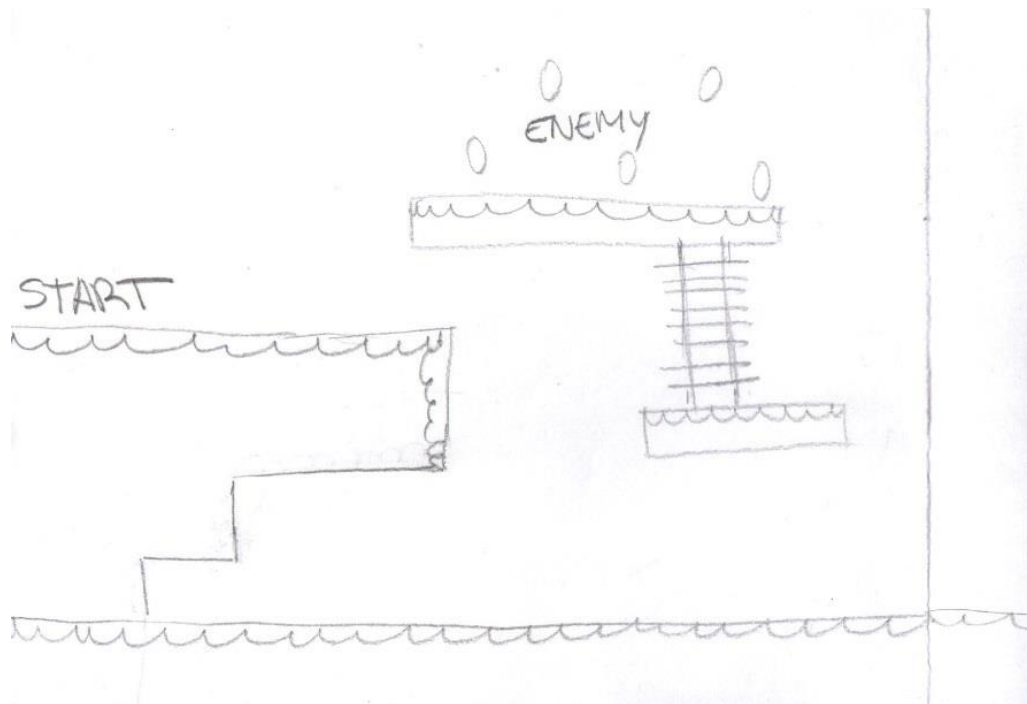


Figure 31: Level 2 Part 1 (self-made)

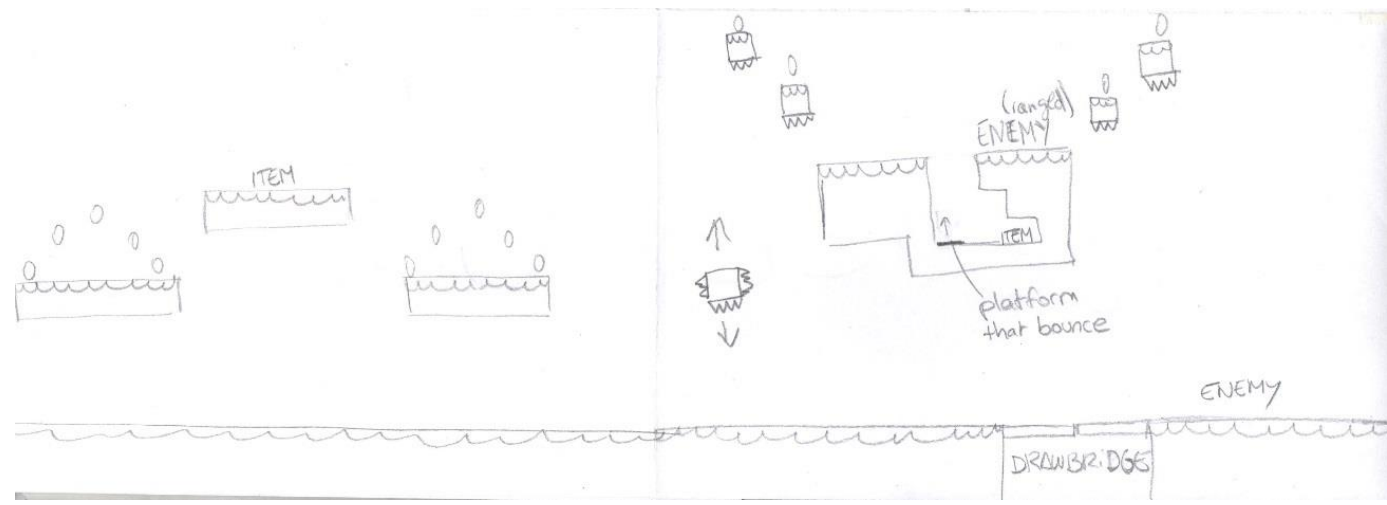


Figure 32: Level 2 Part 2 (self-made)

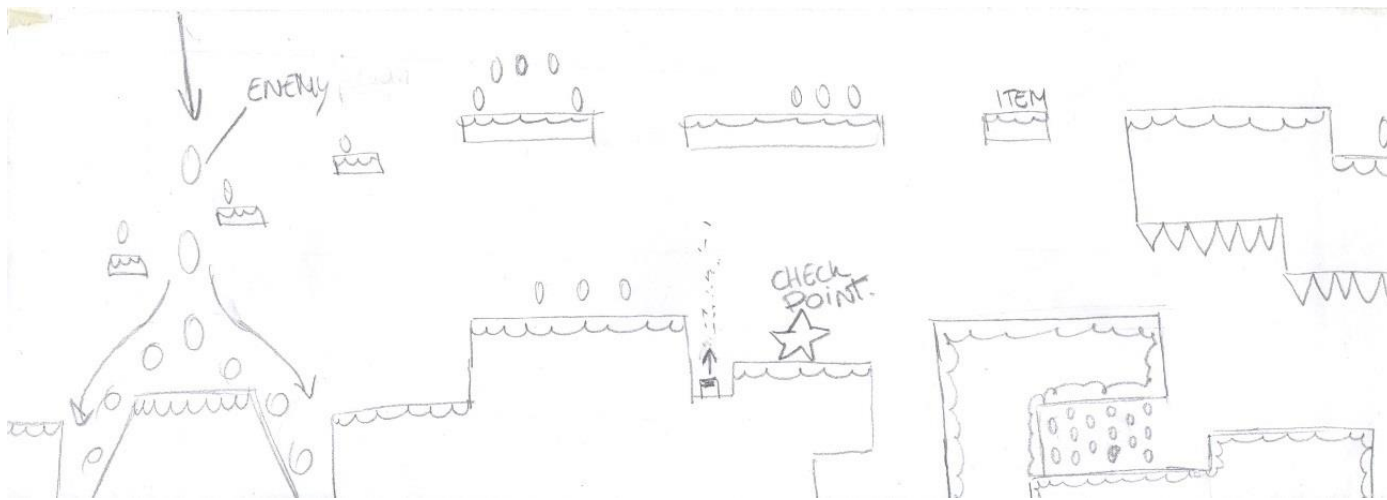


Figure 33: Level 2 Part 3 (self-made)

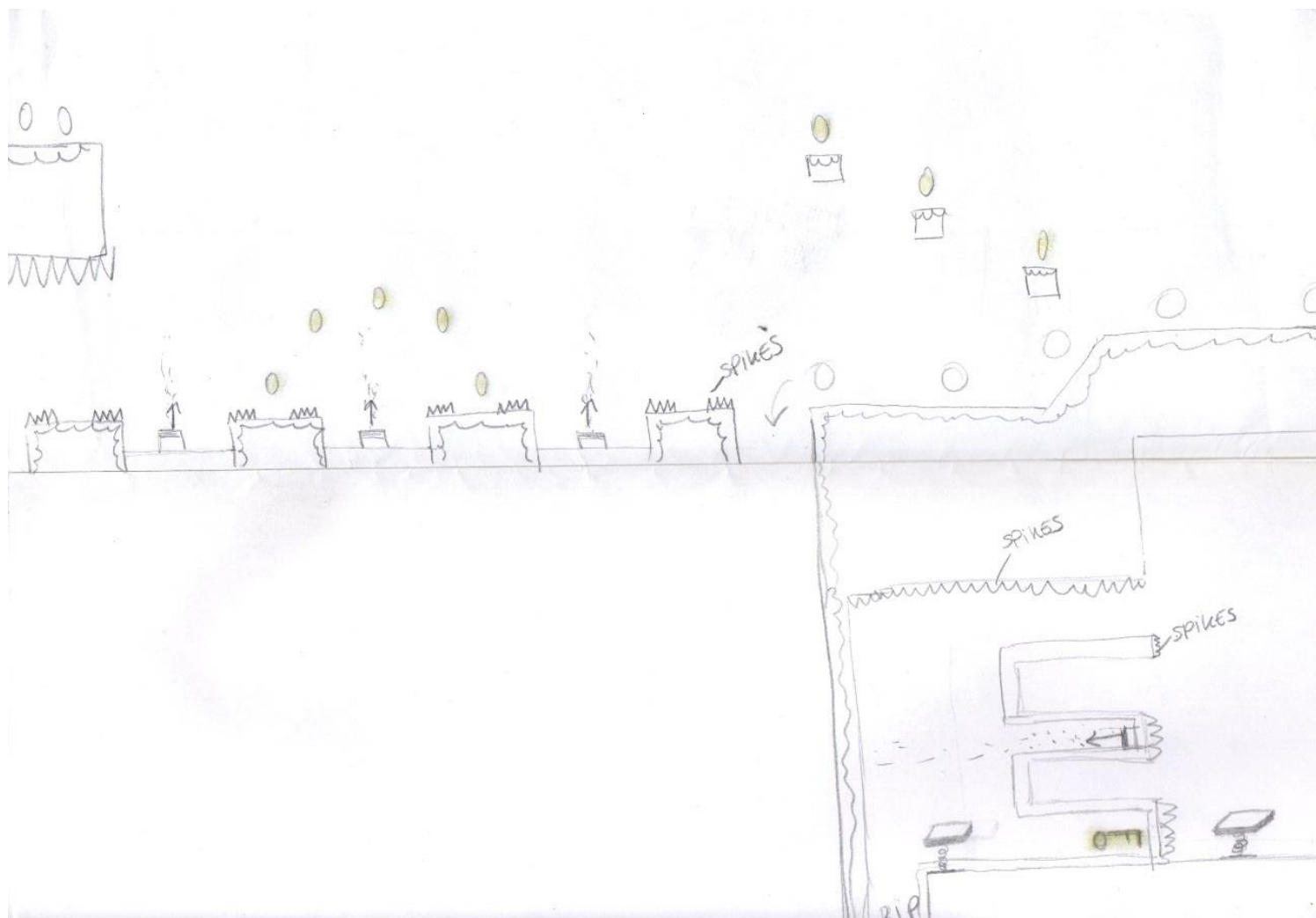


Figure 34: Level 2 Part 4 (self-made)

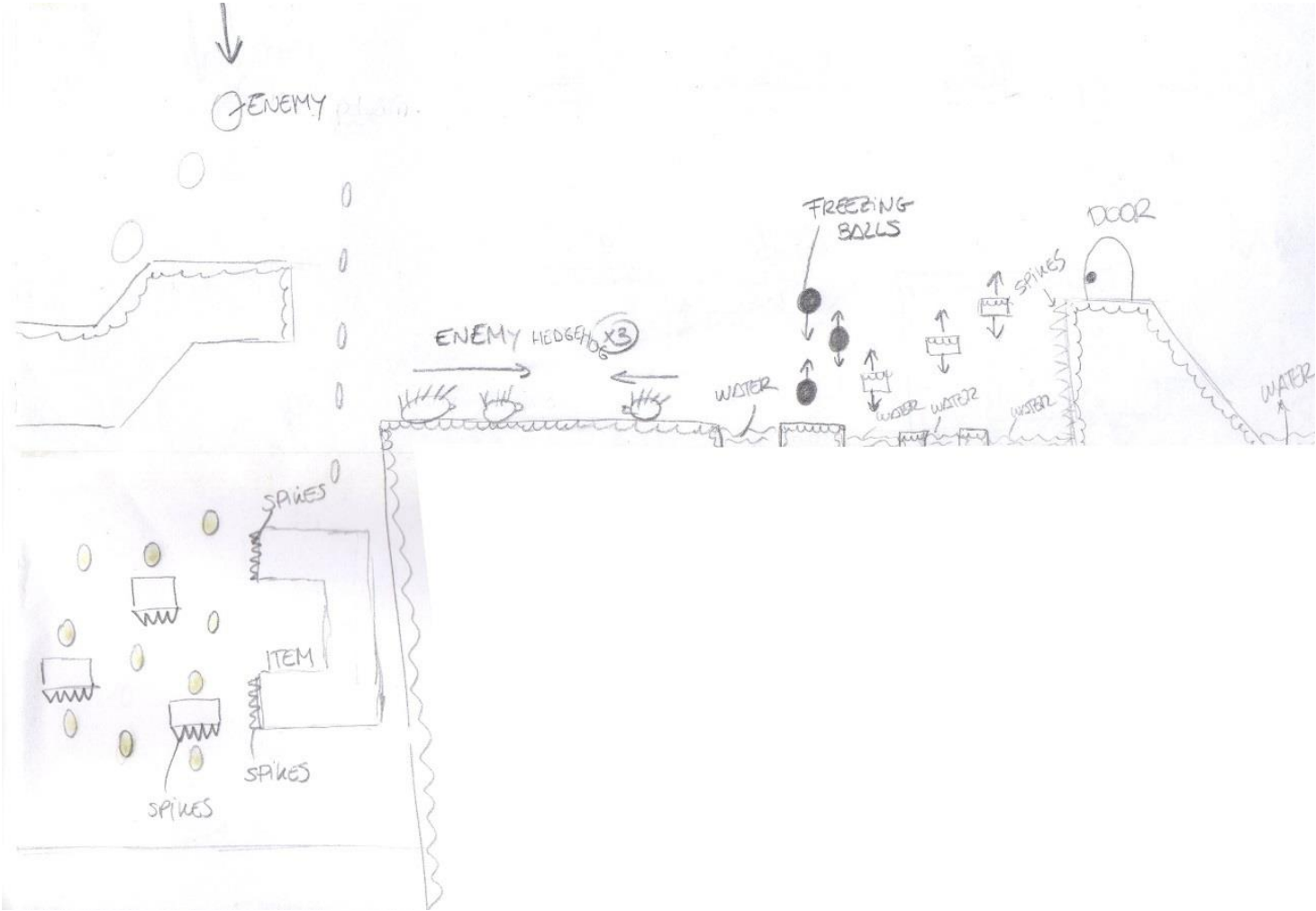


Figure 35: Level 2 Part 5 (self-made)

5.2.6.4 Boss Level 2

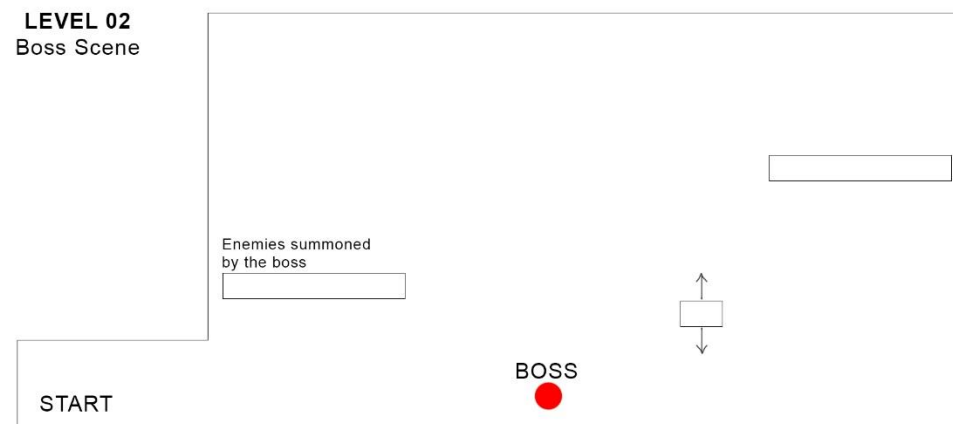


Figure 36: Boss level 2 (self-made)

5.2.6.5 Level 3

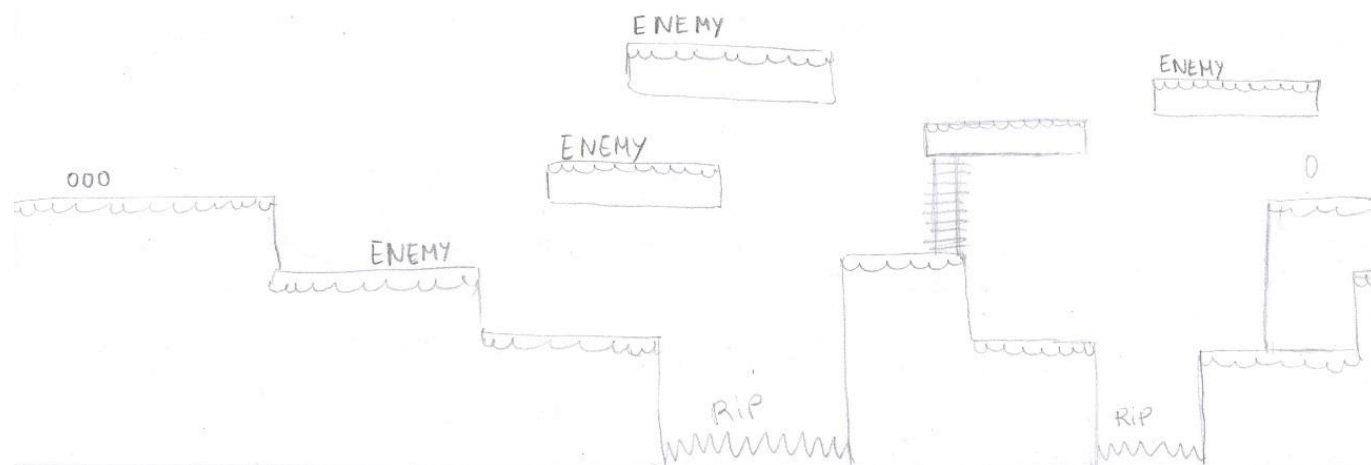


Figure 37: Level 3 Part 1 (self-made)

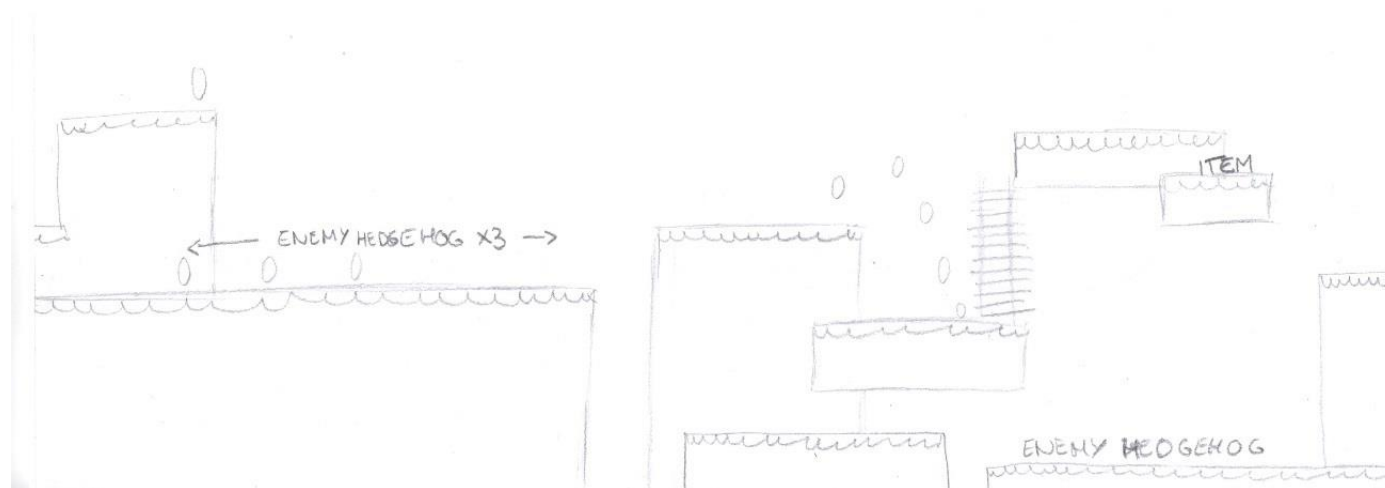


Figure 38: Level 3 Part 2 (self-made)

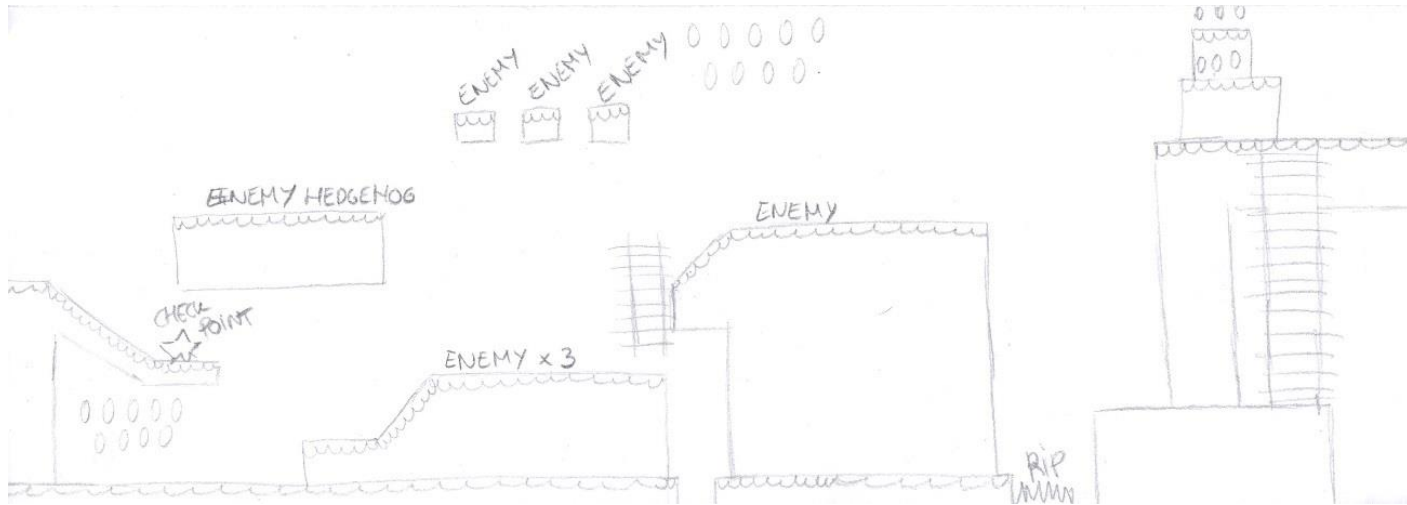


Figure 39: Level 3 Part 3 (self-made)

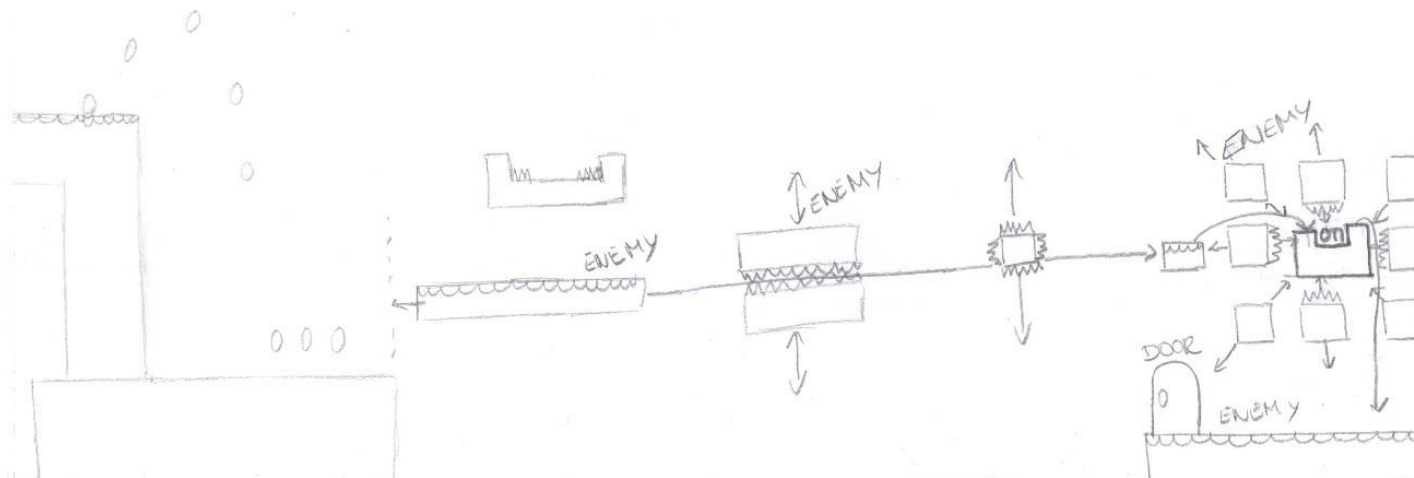


Figure 40: Level 3 Part 4 (self-made)

5.2.6.6 Boss Level 3

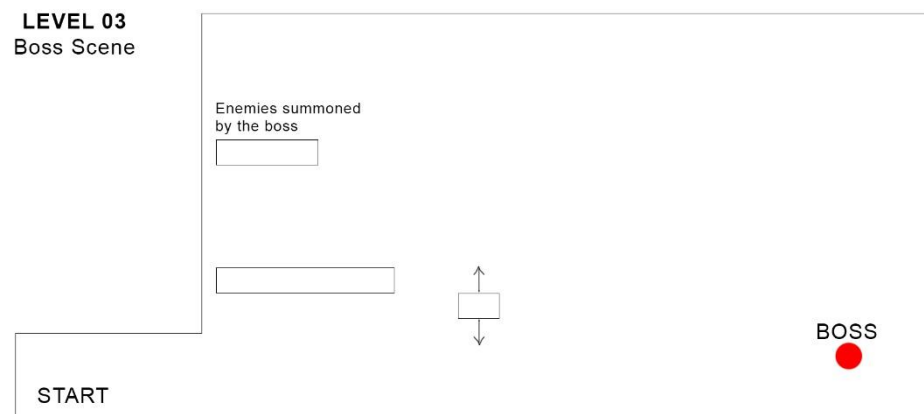


Figure 41: Boss level 3 (self-made)

5.2.7 User interface design

The User interface in this game consist in the HUD (Heads Up Display), the Store, and the character windows.

5.2.7.1 HUD

In the case of the HUD, there are some different attempts before the final design. The main problem was where to put the skills button, so the player can see them without any effort.

Therefore, the following design was discarded, because the buttons on the right are not where the player has his attention and he will have to look for them.

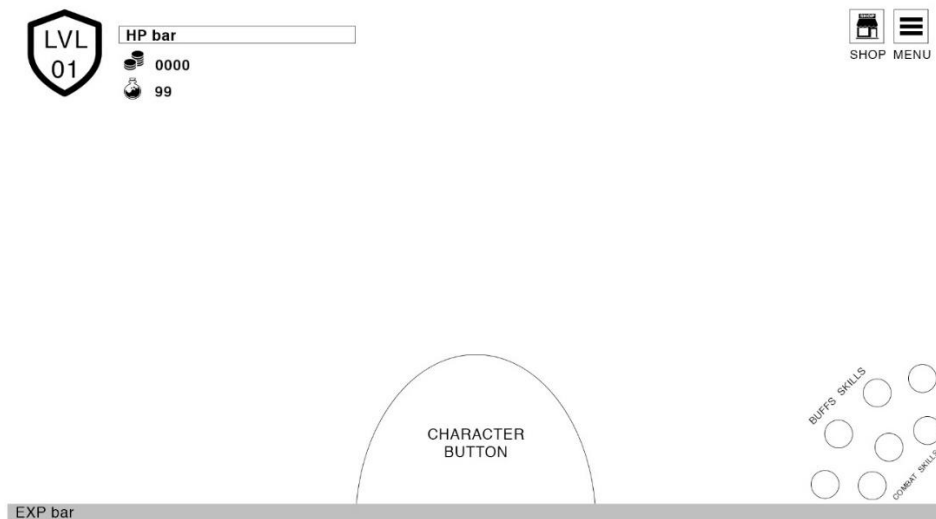


Figure 42: HUD idea (self-made)

Instead of this, put the players on the center of the screen was a good idea, so the final design was the following:

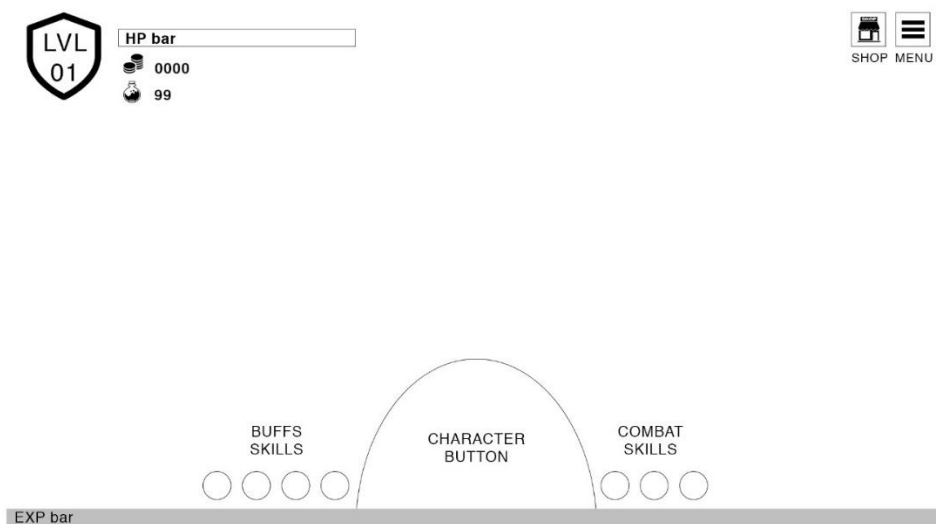


Figure 43: HUD definitive (self-made)

5.2.7.1.1 Skills icons

The following icons are the ones used for the player skills:

Dash

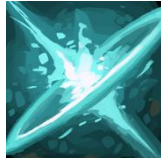


Figure 44: Dash icon (bought asset)

Dagger Throw



Figure 45: Dagger throw icon (bought asset)

Dark shadows

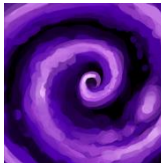


Figure 46: Dark shadows icon (bought asset)

Sharp sword



Figure 47: Sharp sword icon (bought asset)

Extra armor



Figure 48: Extra armor icon (bought asset)

Blood sword



Figure 49: Blood sword icon (bought asset)

Deflection

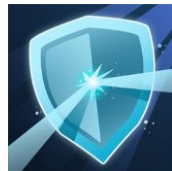


Figure 50: Deflection icon (bought asset)

5.2.7.2 Store

The Store is quite simple, as it is a prototype, and there is no inventory, it has a list of items with an icon, a name, a description with the attributes of the player that will be increased and the cost each item has, including a button to purchase the item.

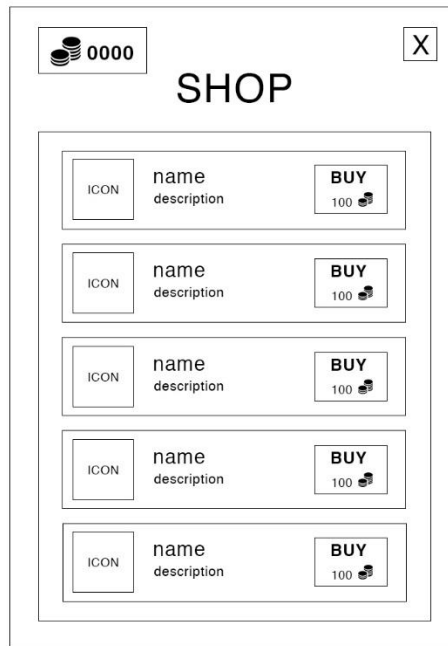


Figure 51: Store design (self-made)

5.2.7.3 Character panel

The character panel is used by the player to check all the attributes he has, and also to upgrade them each time he levels up, using the Skill Points available.

In this case, the design was made on paper and implemented in the game without any change.

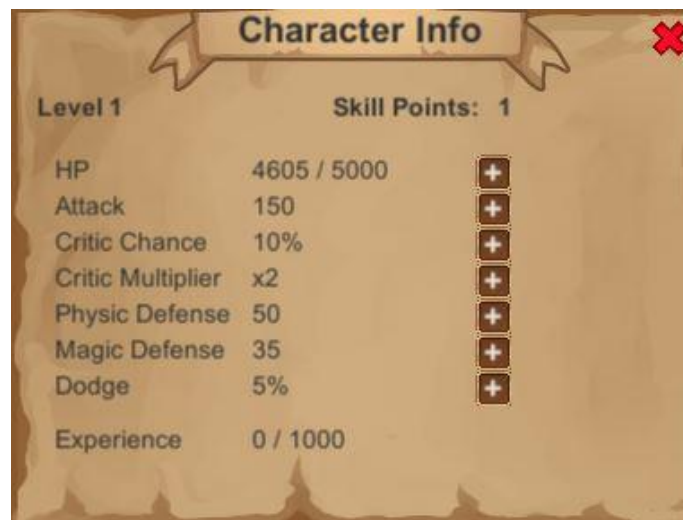


Figure 52: Character panel design (self-made)

5.2.7.4 Menu panel

The menu panel is just for quitting the game if the player wants to and also has two sliders to adjust the volume of the game, one for the general background music and another one for the game effects.

5.2.8 Audio and sound effects

The audios have been selected from a lot of the ones that was on the bought assets package. There are sounds from all different things:

- Background music, which includes:
 - Regular level music
 - Boss's combats music
- Player attack's: One audio if he hits and enemy and another if he does not
- Player skills: throwing the dagger and the "dark shadow" skill
- Player buffs have another special effect sound
- Enemies and player deaths
- User interface sounds when clicking buttons
- Victory on a boss fight
- Collecting items, coins and the key of the level
- Using a health potion
- Player death music and death effect sound
- Enemies death effect sound

5.2.9 Levels tile map sprite sheets

At the beginning of the project the 3 levels that were designed were intended to be from different places, one in the jungle, one in a snowed world, and one using a dark theme. But in the end, all of the levels finished being on the snowed world because there were some missing sprites in all of them and in order to create the less possible art, just the snow tilemap was used.

5.2.9.1 Jungle tilemap

This is the jungle tile map intended to but was never used.



Figure 53: Jungle tilemap sprite sheet (self-made from bought assets)

5.2.9.4 Levels on Unity

Using the level design and the sprites from the snow tilemap sprite sheet, the first map was created on Unity:

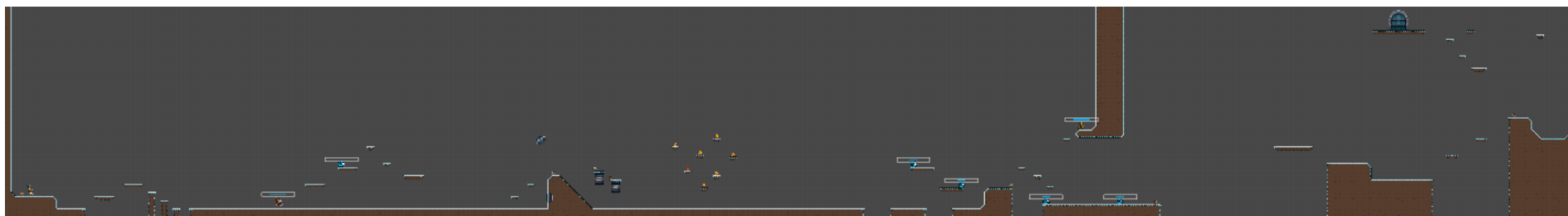


Figure 56: Level 1 on Unity (self-made)

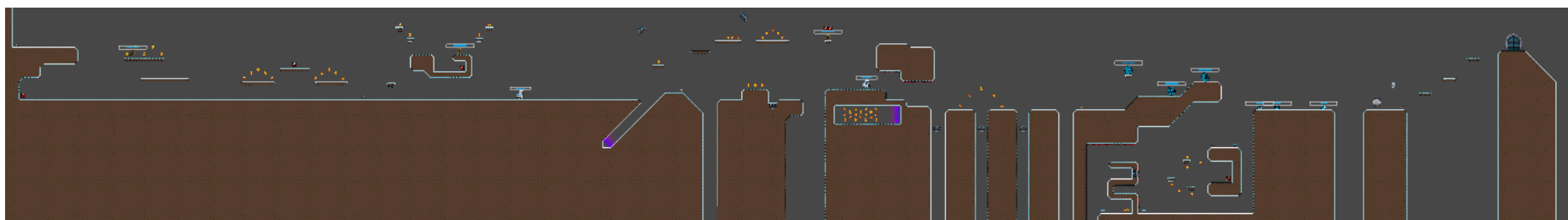


Figure 57: Level 2 on Unity (self-made)

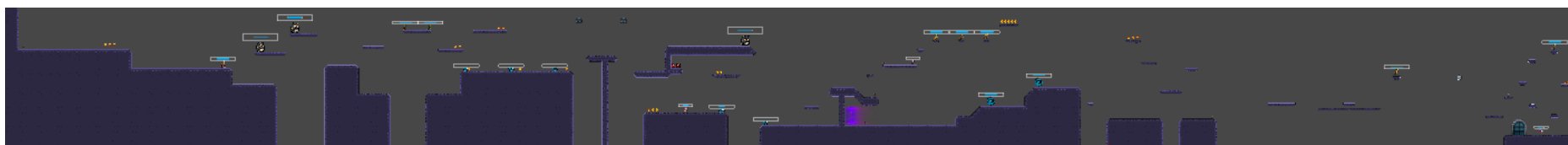


Figure 58: Level 3 on Unity (self-made)

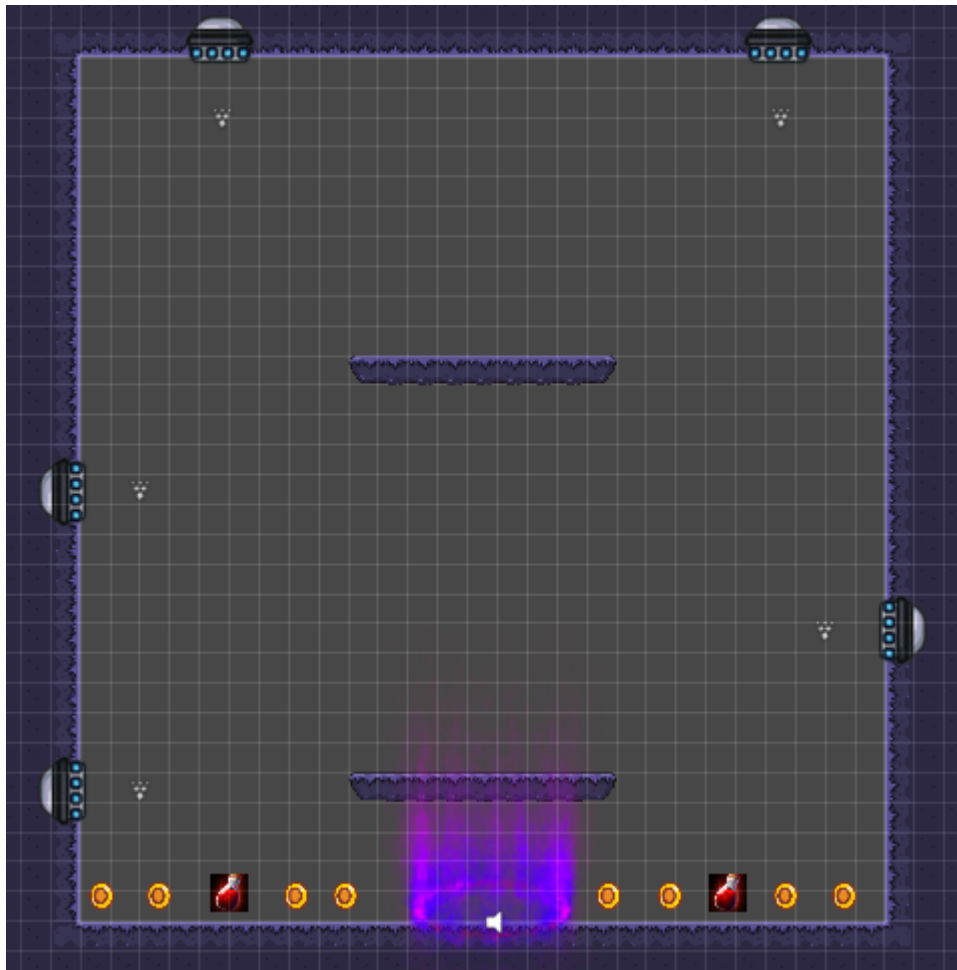


Figure 59: Level 3 secret zone on Unity (self-made)

5.3 Programming

After the design was done, the game inside Unity started having moving characters and so on.

5.3.1 Player

Following the planning, the first step was implementing the player animations, its basic movement, and start creating his skills.

Connecting the player animations and the basic movement take the expected time, although there was an important change on the design mentioned before, it was the point when the acceleration movement was canceled.

The creation of the skills on the other hand took more time than expected, 4 hours and 10 minutes more, to be precise. This delay was caused because, when the skills were working, the feedback given to the player was zero, as the player model lack animations. In order to create this feedback, some particle systems were created to replace those animations.

Due to this delay, environment and world FX's, planned for that week were delayed, as they were not as important as having the player skills on time.

After finishing the skills, on the same week, the skills user interface was created.

The level up system was created on parallel, including his user interface with less time than expected without any major problem.

The player FX's was delayed due to problems with enemies AI and skills, which will be explained in the next part.

5.3.1.1 Player respawning

The player has a respawning system, so if he dies he can come back to life. There are some checkpoints in the game that are invisible to the player. When the player reaches a certain point on the map, the next time he dies he will return to that point.

5.3.2 Enemies

In the case of the enemies, connecting his animations and putting them in the game was done in the expected time.

Once the enemies were in the game with their animations, its artificial intelligence started to be built, thinking, of course, in making it so every enemy can have the same base script.

Because of this, every enemy has a base script with the movement, and each one of them has another one for their attacks and special skills.

As every enemy has different skills, creating all the enemies designed was by far the most expensive task. This caused an important delay, making it not possible to have all the designed enemies yet, but more than enough to test the game and have the needed levels.

5.3.3 Item system

The item system is done all on the user interface. There is the store button where the player has the list of all the items that can be bought.

The first item on the list is an upgrade on the health potion, that can be purchased any amount of times, and each time, the health potion is upgraded and heals more HP to the player when it is used.

The rest of items are equipment that increase the player stats, it is ordered by the type of equipment and the stats it give to the player. Each time the purchase button is clicked, there is a function that is called and, if the player has enough coins to buy the item, they are removed from the player and then, all the stats from the bought item are permanently added to the player.

5.3.4 Levels

The programming on the levels was not a big amount of work because the levels their interactions were not really difficult, except for the ice lance, which take the major part of the time.

- **Moving platforms**

Creating the platform move in a path was really simple, however it had a little difficulty: Making the player move together with it. The solution proposed and used was to make the player game object a child of the platform while he was on top of it; and at the moment he leaves the platform, make the player have no parent game object again.

This caused a bug with the player game object not being saved between scenes that had been solved by using another call on the Unity function: `DoNotDestroyOnLoad(GameObject Player)` when the level changes, which basically makes the game object "Player" not be destroyed when another scene is loaded.

- **Ice lance**

The Ice lance was a bit more difficult than the rest because it was not a sprite like the rest of levels interactions, it was a particle effect created by Unity from a free asset pack. Creating the interaction between a particle effect and the player was a bit more complex because I never had done that, but it ended up working fine and it even did not create any bug afterwards.

- **Coins**

The coins were simply a collider with the animation of the coin, that when the player touched it, it activated a function, so the player collected the coin.

- **Health potions**

The health potions have the same functionality from the coins, except the player gets a health potion instead of a coin, of course.

- **Elastic springs**

The elastic spring consist in a collider that when touched add a force to the rigid body of the player in the Y direction.

- **Spikes**

The spikes have a collider that when the player touches they deal an amount of damage to him and also push him in the opposite direction.

5.3.5 User Interface

The user interface changed a little bit from the start. The design of the image had some variations, so things changed a bit, this caused an unexpected delay at finishing it. However, on the other side, programming it was faster than it was thought at the start, so at the end, the time spent in the user interface was the expected.

A script was created so the panels of the interface can be dragged anywhere on the screen and it is more user friendly.

5.3.5.1 HUD

This includes some different points:

- The enemies and player hit points bars, which are a bar that it is filled in proportion of the current health divided by the total health of every enemy or the player. It is updated every time the character receives damage.
- Coins and health potions accumulated.
- Experience bar; it has the same functioning that the hit points bars but using the player experience and being updated every time the player kills an enemy or levels up instead of receiving damage.
- Skill buttons; those are working if you click on them and the skill can be used, it is used; however, the cooldown reflection on those buttons (for example, having the number of seconds that are needed to be used again). This consist in an image on the top of the button that is unfilled in radial direction following the cooldown of the skill.
- Character information, store and menu buttons. Each one of them opens their respective panels that are explained next.

5.3.5.2 Character panel

The panel was created in the expected time with no problems. It shows all the information about the player and, if there are skill points available, the upgrade buttons appear, and the points can be used to upgrade an attribute.

5.3.5.3 Store

The UI of the store consist in a slider panel created by unity and used the default sprites from the game for the interface, combined with the icons of each item that can be bought.

5.3.5.4 Settings

The settings volume sliders change the music volume and all the sound effects in the game.

All those sounds effects are in a List of audios that are contained in the Audio Manager script.

The “Quit game” button closes the game.

5.3.6 Scene management

As told in the modification of the initial planning, this part was not planned at the start, and moreover, it was more difficult than expected because the most difficult thing on this subject done before the project was to simply change between scenes, but without saving anything from a scene to another.

For this reason, the part of changing the scene was really easy, but there were some bugs with saving the needed elements between scenes.

Saving some of the data was not complicated, like saving the player stats, or the HUD; however, the problem became with the enemies and the player not hitting each other, as they were created in a testing scene and just worked in there.

To solve this, there were needed to create enemies' prefabs and layers of collisions for them, which also were needed to improve the performance of the game. This performance is increased because the matrix of collisions from Unity was modified and it makes some object not calculate collisions with some others, or in other words, make the game objects from a layer only collide with certain other game objects. For example, enemies do only check collisions with the floor and the player, but not between them or between collectables.

5.4 Testing and bugs

It is impossible to do all the coding and expect it to be fine, for this reason, every time a feature was implemented there was a little testing of that feature, checking for bugs.

At the start of them, some were solved just when developing the feature and were not listed, but after the project advanced and more bugs appeared, it was needed to list them.

This list is classified in 3 kinds of bugs depending on their importance:

- Critical: it must be solved because it can make the game unplayable. For example, there was one time that the normal attack from the player did not hit the enemies.
- High: The game can be played, but it modifies the gameplay. For example, with the implementation of the player dash, there was a point that the player could move too fast and pass the level in a few seconds.
- Low: It is an esthetic error that does not affect the gameplay. Some of those may be confused with unnecessary as this is a prototype, but it could lead to be seen as an error in some cases. For example, the arrows from the archers did not rotate.

They also have a status to keep track on them, if they are already solved or not. This is done because if the bug reappears in the future there is a reference on how it was solved.

Bug	Description	STATUS	PRIORITY
1	Player normal attack not damaging enemies	SOLVED	CRITIC
2	Platformer number 11 (level 01) not reachable	SOLVED	CRITIC
3	Dash: Player can cross walls	SOLVED	CRITIC
4	Fireball traspasing floor	SOLVED	HIGH
5	Boss Door is a chest and players not understand it	SOLVED	HIGH
6	Player Dash: Can get way much acceleration after dash	SOLVED	HIGH
7	Hedgehog shoot is not a spike	SOLVED	LOW
8	Archer moving up while attacking	NOT SOLVED	HIGH
9	Controls are a bit uncomfortable	NOT SOLVED	HIGH
10	Skills Icons not correct	NOT SOLVED	HIGH
11	Monk freezing balls should reduce the player speed	NOT SOLVED	HIGH
12	Respawning with speed	NOT SOLVED	HIGH
13	Not enough feedback from hits (Damage number a UI?)	NOT SOLVED	HIGH
14	Sometimes player respawns more than once when he dies	NOT SOLVED	LOW
15	Respawn Point Effect not showing up (Z problem)	NOT SOLVED	LOW
16	Hit floor skill: If player is looking right it traspases the floor	NOT SOLVED	LOW
17	Archer arrow do not rotate	NOT SOLVED	LOW
18	Dash: Player have to look in the direction of the dash	NOT SOLVED	LOW
19	On falling camera should stay idle	NOT SOLVED	LOW

Table 18: Bugs list

5.4.1 Personal and close people playtesting's

With the first playtesting's, mainly done by me, there were less bugs in combat than there was expected. However, after adding almost every enemy that was planned and making more playtesting's, the number of bugs raised fast, creating a quite extended bug list and having to dedicate more time than expected into them.

5.4.2 Post personal and close people testing bugs

The most difficult to find bugs included some audios that were not destroyed when they should, and some enemies not being hit when the player attacked with his normal attack. This last one appeared three different times on different builds, but it could be removed at the end.

Here is the list of the bugs found with them:

Bug	Description	STATUS	PRIORITY	
20	Player jumping something goes wrong with the dash or with a platform	SOLVED	HIGH	
21	Starting the boss level. While loading the player traspassed the wall	SOLVED	CRITIC	
22	Little Yeti moving left makes his face disappear	NOT SOLVED	LOW	
23	Gorilla not working as expected	SOLVED	HIGH	
24	Skills Icons are too small	SOLVED	HIGH	
25	FireBall and Ice Lances Spawner should have a sprite to know where they spawn from	SOLVED	HIGH	
26	Level Up audio is too low	SOLVED	LOW	
27	[Store] Health Potion should be able to be bought any amount of times	SOLVED	HIGH	
28	Teleporting should have a visual effect	SOLVED	HIGH	
29	Some Spikes are not dealing damage	SOLVED	HIGH	
30	Spikes are almost no visible for the player	SOLVED	HIGH	
31	Knife follows the player jumps	SOLVED	HIGH	
32	Some ice lances are invisible	SOLVED	HIGH	
33	Fluffy monster explosion is behind the map	SOLVED	LOW	
34	Fluffy monster HP bar remains in the map after his death	SOLVED	HIGH	
35	Fluffy monster explosion is infinite some times	SOLVED	HIGH	
36	Buff skills noty working well the second time they are activated	SOLVED	HIGH	
37	Not enough feedback from buff skills	SOLVED	HIGH	
38	HP from player not showing actual HP until those are changed	SOLVED	LOW	
39	Player respawn effect is displaced	SOLVED	LOW	
40	Player respawn: Player should not be able to move until the effect ends	NOT SOLVED	LOW	
41	Player normal attack does not feel right	SOLVED	HIGH	
42	Some enemies are not hit by some attacks	SOLVED	CRITIC	DUPLICATED
43	Hedgehogs get stuck sometimes	SOLVED	HIGH	
44	Reflect Buff does not work	SOLVED	HIGH	
45	Hedgehog top spikes are shoot with the sprite rotated	NOT SOLVED	LOW	
46	In some parts the floor takes too much screen space	NOT SOLVED	LOW	
47	Skill icons too small	NOT SOLVED	HIGH	DUPLICATED
48	Some enemies are not hit by basic attack	SOLVED	CRITIC	DUPLICATED
49	Audios not removed when gameobjects are deleted (could crash?)	SOLVED	CRITIC	
50	Boss Level 02 - Starting Gorilla jump hits the roof	NOT SOLVED	HIGH	
51	Gorilla Jumping Attack damage radius is too high	NOT SOLVED	HIGH	

Table 19: Personal and close people post-testing's list

5.4.3 Post personal and close people testing changes

After the first playtesting's from myself and some close friends, before the external playtesting, there had been several changes to the game in order to make it better. Those are the following.

5.4.3.1 Little Yeti and Ice Gorilla design

The jump was decided to be of random height, so the player can't know exactly where he will fall off. On the initial state the player could be used to the jump and if he knew exactly where he will fall, it was so easy to dodge, making it random in a range solves this and makes the combat with those enemies, specially the Ice Gorilla because it is a longer fight, more interesting.

5.4.3.2 Level design change and new sprites

The initial idea of making freezing balls on the second level was replaced with a moving rock with spikes around. That was because the freezing balls just let you there, static for a moment and if no enemies were around it was just annoying. Instead, some new sprite was created.



Figure 60: Rock with spikes sprite (self-made)

In the design of the second level there were also platforms and floor with spikes, so those new sprites were created as well.



Figure 61: Platform with spikes sprite 1 (self-made)



Figure 62: Platform with spikes sprite 2 (self-made)



Figure 63: Floor with spikes sprite (self-made)

After adding these spikes, there was a problem of visibility with the ones attached on the floor (Figure 60): they were difficult to see, so to solve this, they were painted in red.

5.4.3.3 New level 2 zone

In order to give a bit more importance to the exploration, a little secret zone with a lot of coins was added into the map. This secret zone is visible to the player as he advances, but he will need to find out how to enter this zone.

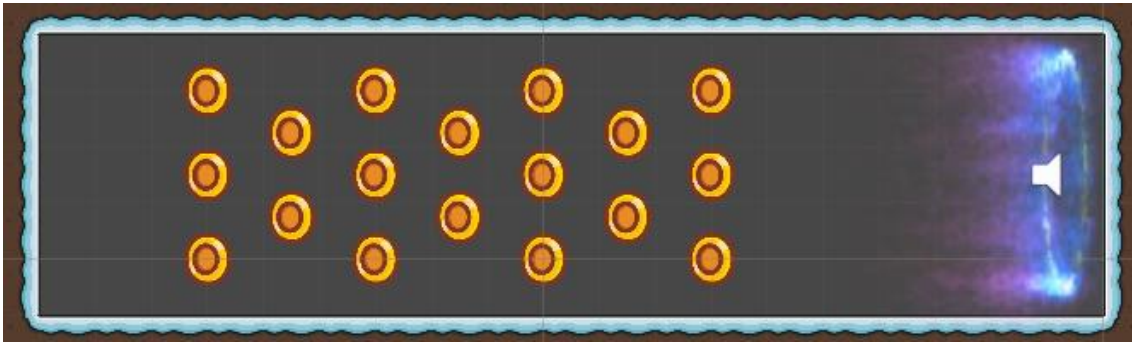


Figure 64: Secret coins zone level 2 (self-made)

To access this zone, the player must fall from a place that at first it seems to be falling to death, but that in the end of this zone there is a teleporter that brings you there.

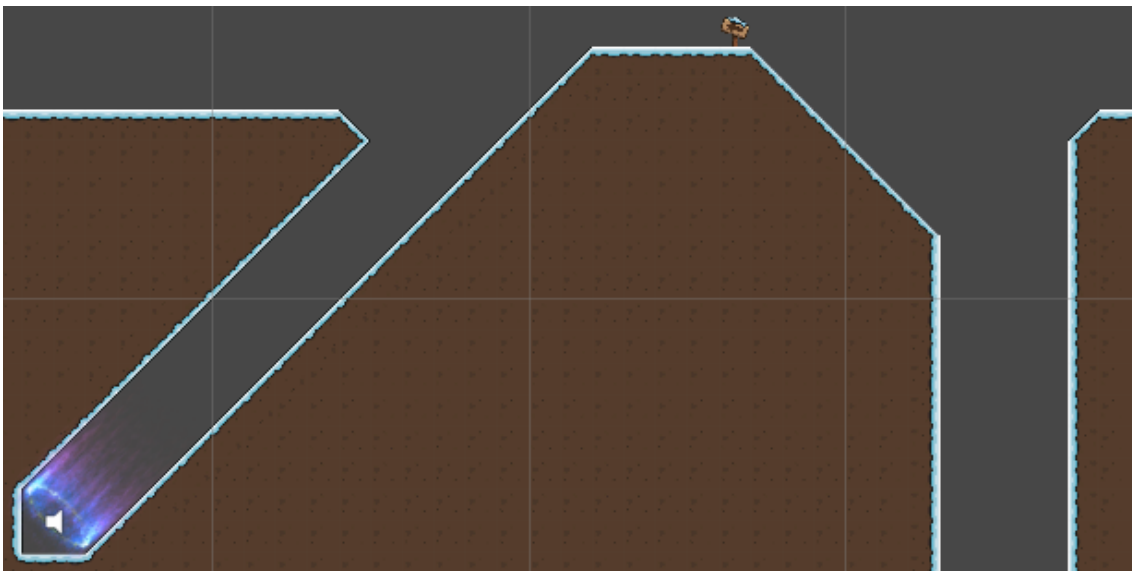


Figure 65: Secret teleport zone level 2 (self-made)

5.4.3.4 New particle effects

A bunch of new particle effects were added because there were way too many things that were not giving enough (or even none) feedback to the player. Those new particle effects were looked for and found on a package from the Asset Store and a free source asset from a guy called CodeManu:

- <https://assetstore.unity.com/packages/vfx/particles/cartoon-fx-free-109565>
- <https://codemanu.itch.io/>

5.4.3.4.1 Player buffs activation

The feedback from activating the buffs was only that the character stats were added in terms of numbers. To give enough feedback there was some particle effects added around the player activated while those buffs were also active.



Figure 66: Reflect damage skill effect

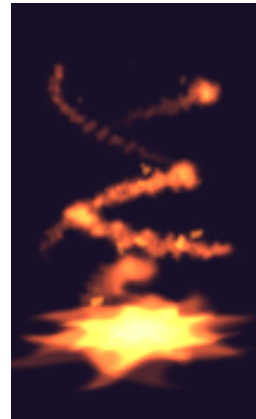


Figure 67: Increase attack skill effect

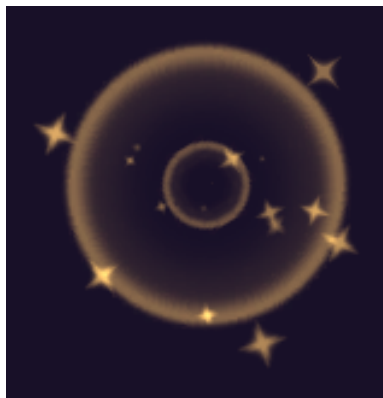


Figure 68: Increase defense skill effect



Figure 69: Blood sword skill effect

5.4.3.4.2 Teleport

After adding the teleporter to the zone without anything more than a trigger, it was really strange for the player without a visual effect, so there was also a particle effect used for this.

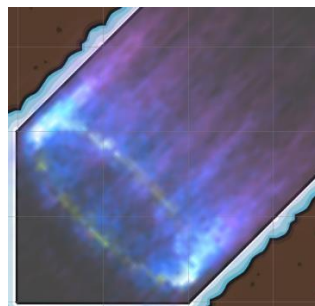


Figure 70: Teleporter effect

It also is used as a clue to the player on how you may enter the secret zone (looking into an identical particle effect).

5.4.3.4.3 Summon enemies

When the enemy bosses activated their summoning skill, it was also really strange that an enemy just appeared on the map, with a little summoning particle effect this was solved and looked really better.

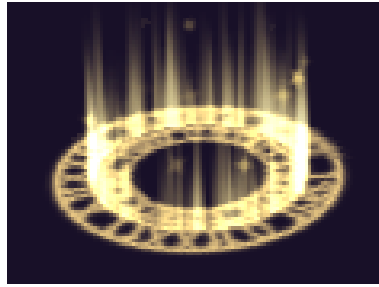


Figure 71: Summoning effect

5.4.3.4.4 Ice lances and fire balls spawners

The spawners from both ice lances and fire balls had the same problem that summoning enemies, they just spawned there with no apparent reason, so on each one of those spawners was added a constant particle effect so the player knows where they came from.

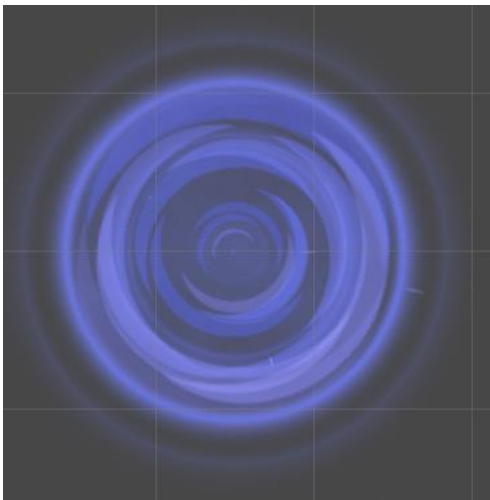


Figure 72: Ice lance spawner effect

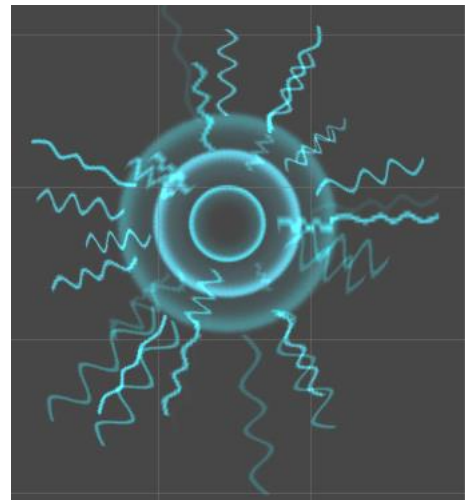


Figure 73: Fire ball spawner effect

5.4.3.4.5 Dark Shadows skill

The Dark Shadows skill from the player already had a particle effect, however, while searching for the others that really were needed, a new one that seemed way better was found, so it was replaced.

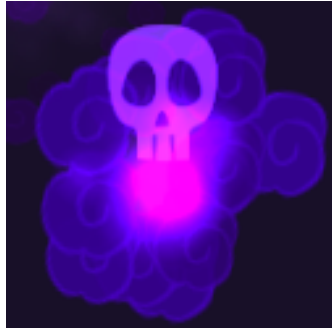


Figure 74: Dark shadows skill effect

5.4.3.4.6 Player respawn

The previous used player respawn effect was not really good, so it became replaced by this one found on the same pack.

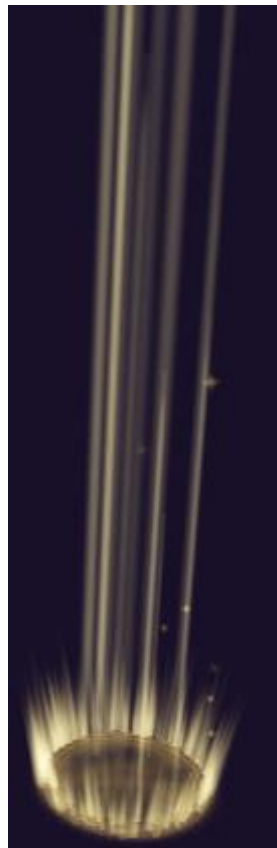


Figure 75: Player respawn effect

5.4.3.5 New audio effects

Together with the new particle effects, and both for giving feedback to the player and making the game better some audio effects were added to the ones the game already had:

- Summoning an enemy
- Opening a door to enter the next boss or level

- Monk teleport skill and player teleport
- Victory sting on killing a boss

5.4.3.6 Player guidance

There was also detected that in some cases players got lost on where the key may be. Although this was not the majority of the cases, when this happens the game becomes a bit frustrating, so in order to improve a bit more the player guidance on the levels, there were added some signs on the map to give directions and even warn him from a coming danger.



Figure 76: Direction sign sprite (bought asset)



Figure 77: Danger sign sprite (bought asset)

5.4.3.7 Player buffs design

Player buffs that increased the attack and the defense of the player were modified, instead of giving just an amount of attack and defense, it was decided to change that to give that amount plus a percentage of the player current attack or defense respectively. This decision was made because on the later levels of the player, just giving an amount made those skills useless.

On the case of the defense buff there was another change as well, instead of giving just physical defense, it also gives magic defense. This was made because if the player was fighting against a magic type enemy this skill became useless as well, and some players did not understand why it did not work.

5.4.3.8 Second boss level redesign

The boss level that was initially designed was changed quite a lot because there was way much space on the map than needed, which caused the fight to be less intense than it should be. For this reason, the new map had the same elements the original had, but in a reduced space.

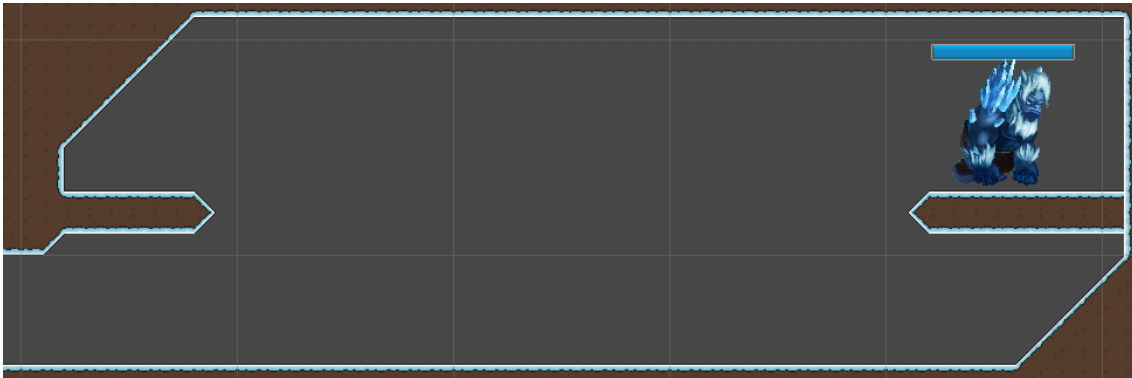


Figure 78: Level boss 02 redesign (self-made)

5.4.4 First external playtesting

Once the game prototype bugs are reduced a lot, after the personal and close people playtesting's, it was time to prepare an external playtesting. For this we only need to create a release and a Google Form with a questionnaire for the testers to answer.

The release is done the same way than the previous ones made by the other playtesting's. However, in this case a test on how balanced the game is, was wanted to be made. For this reason, previous to the release a first balancing process had to be made. This part is explained on the part "5.5.1 First balancing" of this memory.

On the other hand, the questionnaire was not done before and it was important to think in which questions you should ask that can be useful to your game, or in other words, which questions should be made that its answer can have changes to make the game better and which changes they will be depending on the tester's answers.

Based on that, the formulary is divided in three sections.

The first one is about the tester, we need to know if our target is really the one who we expect: young people that plays videogames usually as a hobby.

That's why the first two questions are the tester's age and if he plays videogames often. If those answers coincide with what we expect, this testing will be useful, if not, it could be that it is not useful, and we may not count on those.

The second one is about the game itself, it was important to know the difficulty of the videogame, to see the degree on which the balance that has been done is good or bad. In order to know this, we ask the following:

- If the tester reached the first boss
- If he beat him
- If he reached the second boss
- If he beat him
- If he/she found the prototype easy or hard from 1 to 10, meaning that 5 would be an adequate level.

Also, in order to know if the user interface is predictable enough, we ask if the tester bought anything from the store and if he used his skill points. If he did not know about one of those there will be needed a redesign of the user interface. This can also help in order to know if the results on the difficulty of the level were accurate or not: if a tester found the game hard but did not bought any item and did not use any skill point, it is obvious that the game will be really hard, and then we cannot consider the difficulty results.

Finally, the last part is about the flaws and the strengths of the game. Those questions are for the tester to write whatever they want to, and they include the following:

- Which things did they liked
- Which things did they disliked
- How they will fix the things they did not like
- If they got lost on any point of the game and where
- If they could change anything on the game, what would it be

At the end of the form there is a part to rate the prototype from 1 to 10, so we also will know if the game is fun to play or not.

Playtesting form link: <https://forms.gle/KGY2vAGbHKY4TxNc7>

Playtesting release link:

https://drive.google.com/open?id=1_MTaTr4O6rNBa8_tceZoO24XI9wOUTTD

5.4.5 First external playtesting results

A total of 15 users play tested this first version. The results were pretty similar and most of them point out the same things, which means that the playtesting was done correctly in order to improve the game, some bugs they reported and other ideas helped to make the prototype better.

The age of the play testers was in a correct range between 16 and 26 years old.

Age

15 responses

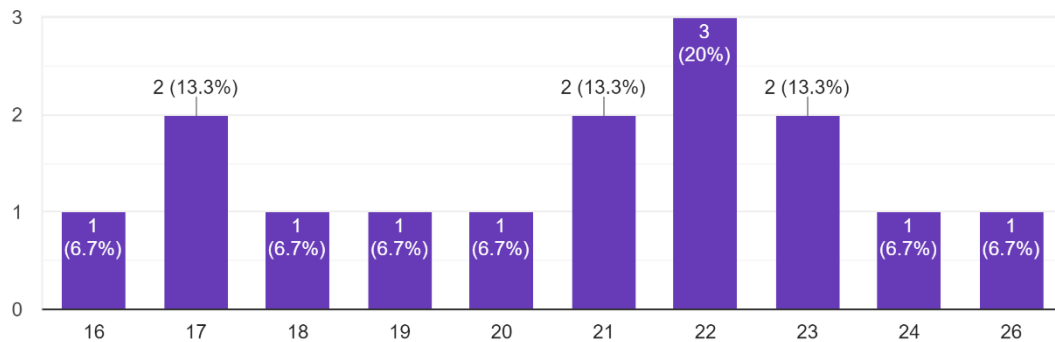


Table 20: First external playtesting (Age)

All of them except one usually plays video games as well, meaning that the results were as expected yet and that the playtest can be considered for improvement.

Every one of them except one beat the first level and reached the first boss, however the problem came when only 9 of them beat him, indicating that it was a bit hard.

After this, all of the ones that beat the first boss got to the second one, and all of them except one beat the second and final boss on this release. Note that the second boss is not the final boss, but the third level was being created when this testing took place.

In conclusion, the game was a bit hard for most of the testers, but not for everyone, which lead to lowering a bit the difficulty but not so much. Those are the results, on the formulary there were instructions that indicated that 1 was too easy, 5 correctly balanced and 10 too hard.

Was the game too easy or too hard?

15 responses

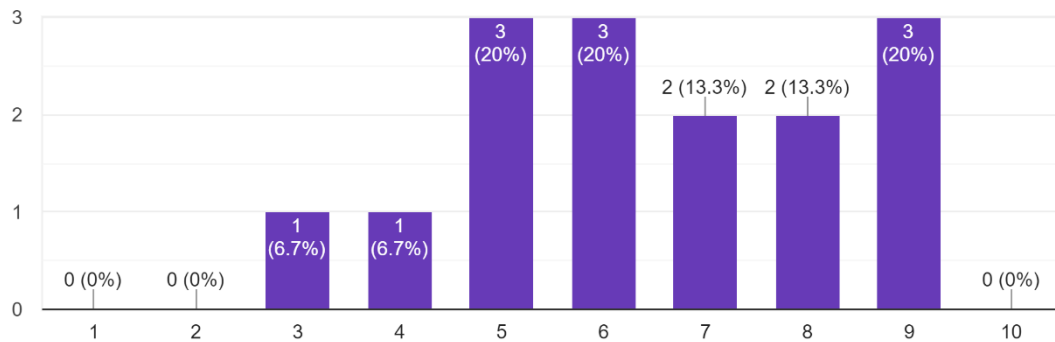


Table 21: First external playtesting (Difficulty)

On the open part of the formulary, the main and most repeated feedback was the following.

On the positive things, they liked the most:

- The movement, they liked the freedom of movement and the control you have of your character
- The platforming, which also relates a bit on the player control
- The aesthetics, which was actually a surprise because they can be hugely improved, as this is a prototype and there are some obvious errors. They were aware that it was a prototype, but it was a bit positive surprising anyway
- Different parts of the combat (player skills, different enemies)

On the negative things, they disliked the most:

- A bit more of decorative details
- Not enough feedback when they got hit by the enemies
- A lot of them did not notice the existence of the store neither the player skill points, probable reason why most of them found the game harder than it is.
- The controls, most of them said they felt uncomfortable

The things they would have changed:

- Add some decorative elements or a parallax
- Differentiate the levels aesthetically
- Use the mouse to attack because you already use it for aiming the dash and the dagger
- Put the skills description on the game
- Something to know the enemies hit the player
- Reset buffs when the player dies

Only one player got lost in some point, particularly they did not find the second key, however if the previous questions are checked, it can be seen that every player that beat the first boss could get to the second, which means that in the end that tester could manage to get the key.

Finally, on the scoring the prototype question, it gets an average score of 6.53 out of 10, which can be considered pretty good, especially because there were a lot of elements that were missing that had been implemented later.

The playtest was also really helpful because a lot of bugs were found because of it:

- When player died sometimes he continued moving after respawning without touching anything
- Sometimes the main menu keeps opening and closing by hitting the spacebar
- The first boss teleports to top and when you get there he teleports back down again

5.4.6 First external playtesting evaluation

The first qualification that should be done to this playtesting should be astonishingly useful. A lot of improvements and bugs were made thanks to it and the prototype quality was hugely improved.

The first thing that had been done while playtesting results were coming were writing down all the bugs and improvements that were reported. In some cases, the solution was already written like in the case of the player controls, while others had to be thought and redesigned.

5.4.7 First external playtesting changes

With all the feedback, the following changes were done.

5.4.7.1 Controls redesign

The controls the playtesting release had were under though, before creating it, as almost every playtest was made by myself, I did not find any problem with them because I was used to it.

Taking advice from them, the mouse buttons took a primary place, and it came out great in the next playtesting's.

Control	Key Pre Playtesting	Key Post Playtesting
Move left	A	A
Move right	D	D
Jump	SPACE	SPACE
Use health potion	S	S
Normal attack	Q	Mouse right click
Dash	W	Shift
Throw dagger	E	Mouse left click
Dark shadows	R	W
Improve attack buff	1	1
Improve defense buff	2	2
Reflect damage buff	3	3
Life steal buff	4	4

Table 22: Controls inputs

5.4.7.2 Damage popup

The worst and more repeated problem on the playtesting was that the player did not know when he hit or got hit by enemies. In order to give feedback to the player about that, it was decided to create damage pop ups on the center of the character that was hurt.

This text has the exact amount of damage dealt and moves to the side top of the character before fading out, if it is a regular attack it is orange, if it is a critical or is an attack that deals “true damage”, which means that that attack ignores the defense the character has, the text size is a bit bigger and red.



Figure 79: Damage popup

In the case of the player, this popup also appears when he uses a health potion or gets his health recovered by using the life steal buff. In this case the number appears with a “+” sign before the number and in green color.

5.4.7.3 Skill points and store feedback

Another really problematic issue was that a lot of players did not know about the skill points or the store (or even both of them) until they read about them on the formulary.

Although it was written on a file of the release it was expected that almost none of the players if not none at all would read that, however, there were more players than expected to not realize those features.

To avoid the unknowingness of the store and the player skill point, it was decided to put the following sprite at the border of the buttons to access the panels:



Figure 80: UI alert sprite (bought asset)

However, putting it in the border of the buttons was not enough, so in order to give it relevance, they do not only appear each time the player has any skill point to use available on

the character panel or every time the player has enough coins to buy the most expensive item on the store, but also increase and reduce its size continuously.

5.4.7.4 Change theme each level

At first, having a different theme on each level was not considered important because it was artistic, and it was never the point of this prototype and it was quite an extra work; however, it was decided to add it afterwards because it gave the sensation of progress in the game.

Unfortunately, this change could only be applied on the third and last level for time reasons, so in the end the first two levels were creating using the snow tile map and the last one used the dark themed tile map seen previously on the “5.2.6.2 Dark theme tilemap” point.

To create this, some sprites had to be done and added to the game:



Figure 81: Dark theme spikes platform 1 (self-made)



Figure 82: Dark theme spikes platform 2 (self-made)

5.4.7.5 Armor redesign

There was a big problem that had to be solved on that release: the way armor worked. There was a huge critical bug discovered, if the player only stacked defense and bought all the defensive items, he became immortal for some kind of enemies.

For this reason, instead of just subtracting the defense when any attack came, a percentage of the damage was ignored. This percentage is calculated using the following formula:

$$Damage = Damage * \frac{100}{(100 + Armor)}$$

Creating a logarithmic scale like in this graph:

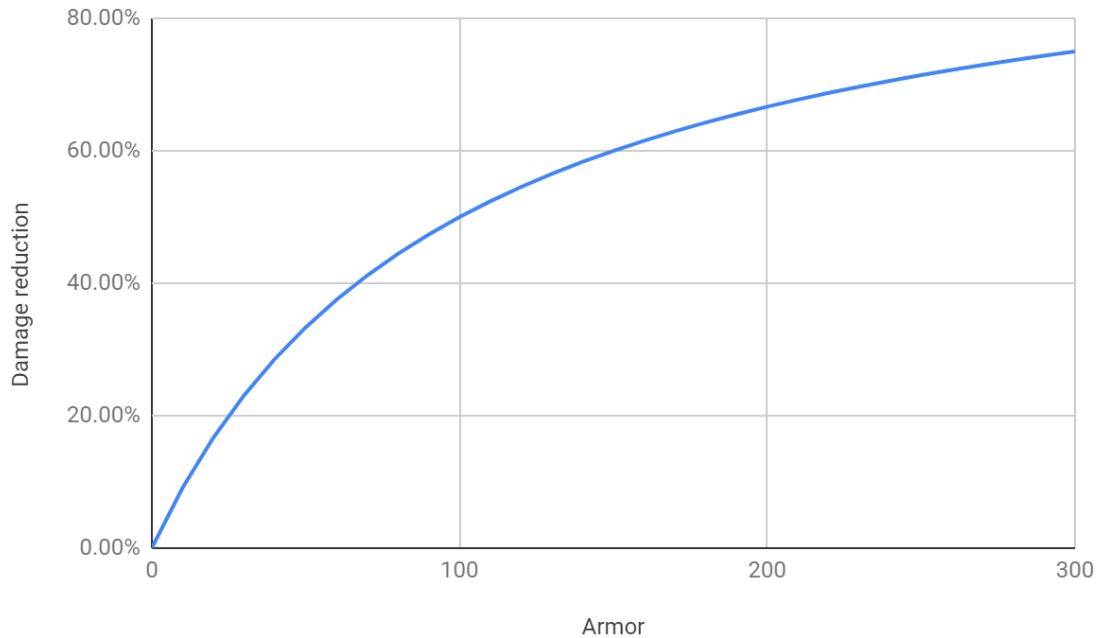


Figure 83: Armor to damage reduction % scale (self-made)

5.4.7.6 First boss level redesign

There were also some changes on the first boss room (Monk), because when you tried to chase him after teleporting, as it was pretty difficult to get to those teleporting points, when you get there, the boss was already teleporting to another position. That's why more platforms that go up and down were added and the speed of those was increased.

5.4.7.7 Buffs reset on player death

Once the player died, all skills cooldown and buffs effect were restored. Solving this caused some little bugs with buff effects resetting twice on death, but all of them were solved as well.

5.4.7.8 Other minor changes

There were also some minor changes that were reported and solved:

- Some enemy projectiles went through the walls
- Teleport particles color changed because it could be confused with ice lancer spawners and fire balls spawners
- Added parallax
- Low the difficulty a bit

5.4.8 Second external playtesting

After the changes made result from the first external playtesting and the implementation of the third and last level, another playtesting was made. This was not planned on the start but

after viewing the positive results and the really good improvements that the first one caused, it was thought it would be really good to do another one in order to improve more things and, moreover, check if the problems the first playtest had were solved in this new version or it was needed to make more changes.

The formulary was exactly the same because it worked fine on the first playtesting and it reported different issues, not only one. The only thing that changes was asking if they reached the third boss and if they defeated them, as this level was on the first version.

Playtesting form link: <https://forms.gle/DVoDNSmTvWNQgWCeA>

Playtesting release link:

https://drive.google.com/open?id=1Qs0_2p5N2gF0J_cWVV99GpE_AjtH3mal

5.4.9 Second external playtesting results

In this second playtesting a total of 10 users tested it. With all the changes made, the results were way better than the previous version.

The age of the play testers were in a correct range between 15 and 26 years old.

Age

10 responses

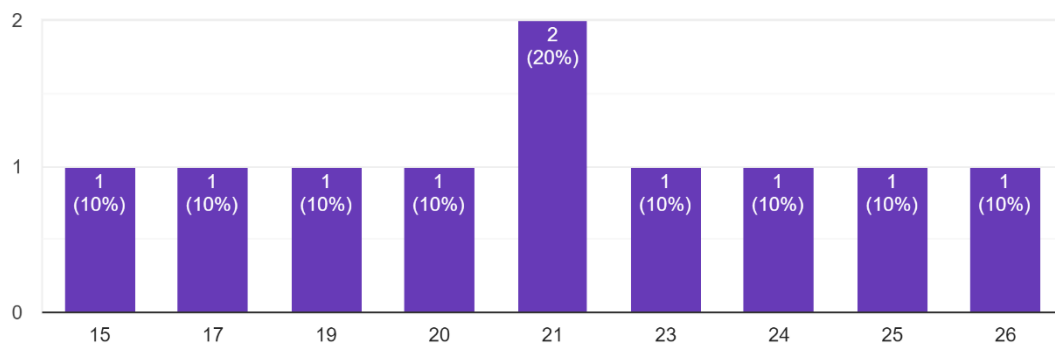


Table 23: Second external playtesting (Age)

All of them usually play videogames and all of them get to the first boss. Just one could not beat him and all of the rest (9 players) reached the second boss.

From those 9 players, 7 beat the second boss and all of those 7 reached the third boss, which was beaten by all of them except one.

9 of the 10 bought items from the store, and 8 of the 10 used their skill points. Meaning that the change that we made in the point “5.4.4.3 Skill points and store feedback” was nicely done and had the expected result.

In this case, although the last balancing was making the game just a bit easier, as expected, the fact that the players bought items on the store and used their skill points make the game easier, maybe even too much. In this case, the difficulty graph was more centered to the 5:

Did you find the game easy or hard?

10 responses

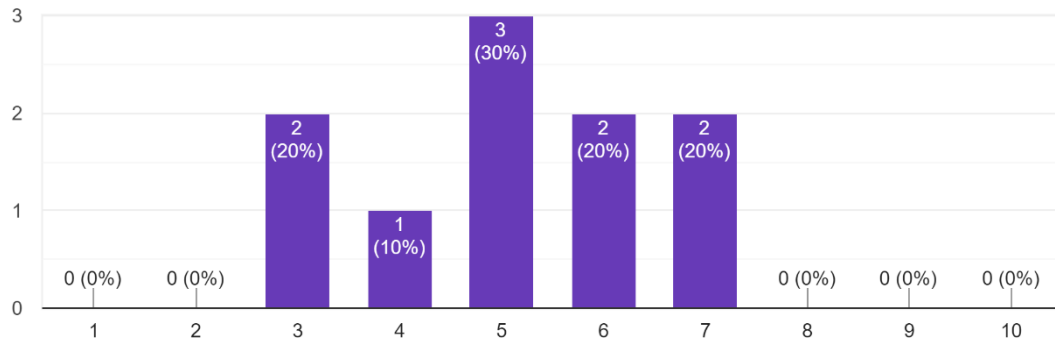


Table 24: Second external playtesting (Difficulty)

On the open part of the formulary, the main and most repeated feedback was the following.

On the positive things, compared to the first playtesting the players seem to like the same things, not much difference in this case, but it is good to know that they still like them. They said they liked the most:

- The combat
- The platforming
- The skills from both player and enemies
- The player mobility

On the negative things, things changed a lot, instead of being general things they are particular details, meaning the solutions that were implemented after the first playtesting worked pretty well. They disliked the most:

- UI skill buttons too small
- The dagger icon cooldown could not be seen clearly
- They did not know what the skills did at the beginning.
- The lack of check points on the third level
- On the fluffy monster they did not understand he will explode the first time they encountered it.

On the things they would have changed, it was not so useful this time, because all the solutions were pretty obvious like in the dagger icon problem, change the background color, or in the not knowing what the skills do, putting a description in each one of them. However, this is a good thing because it seems to be only little problems.

From all 10, just one could not find the second level key like in the first playtesting, but he ended up finding it, so there is no problem at all with that.

On the change or add anything on the game question, this playtesting was completely different, they said to add more skills to the player, or add more audios to the game, add more enemies, add a list of the items bought and include the possibility to crouch, or in other words, to continue the game development adding new features, new enemies and new levels.

Finally, on the scoring the prototype question, it get an average score of 7.1 out of 10, which is even better than the previous one and can be considered a good score.

5.4.10 Second external playtesting evaluation

In this second playtesting, the evaluation can be also considered as useful, but in another sense that the first one. Instead of useful in the sense of that they gave me a lot of bugs and improvements, it as useful to see that the game is actually fun and that not only they suggest improvements to make, but also it can be seen to want more enemies and more skills or in other words, more of the game.

5.4.11 Second external playtesting changes

5.4.11.1 UI buttons changes

From the feedback, two changes were done with the UI, the skill icons were bigger, and the “Throw dagger” skill were given a new background, instead of the old black one, a really clear grey.



Figure 84: Throw dagger new icon (self-made from bought asset)

5.4.11.2 Level 3 checkpoint

On the third level, a new checkpoint had been created in the start of the level, so there is not a lot of distance between every checkpoint.

5.4.11.3 Fluffy monster wick spark

In order to make the fluffy monster to know that he will explode, a particle effect of a spark was created on his wick and follows his rotation.

5.4.11.4 Skills description

The solution found to let the player know what each skill he has does is to create a popup description of each skill when the mouse is over each skill button.



Figure 85: Skill description popup

5.4.12 Unity bug

After the second playtesting a unity error message started appearing on the Unity editor's console. It said the following: "SerializedObject target has been destroyed. UnityEngine.GUIUtility:ProcessEvent(Int32, IntPtr)".

It turned out that this error was from Unity, and some lighting rendering. Luckily, as the game is in 2D, lights are not needed, which was good to know because they were on some particle effects that were added after the first external playtesting and could be removed, making the game calculations simpler and because of it, making it go a bit faster.

The solution in this case was simply delete those lights from the particle effects that had them because they were meant to be used on 3D as well.

5.4.13 Soak idle testing

A soak idle testing has also been made. In order to do it a new release of the game has been created with one enemy of each kind in the starting position of the player and the player had been given a million HP in order to survive some hours without dying.

The test lasted for 8 hours and no problem or bug could be found. After those 8 hours the game could be completed as the normal could.

5.5 Polish and balancing

At the end of the project there is expected to be a polish of everything in general, specially bug solving and a final balancing of the design so the game is not easy, and the player feels boredom when playing and neither too hard so the player can be stressed and cannot finish the game.

This balancing will be done using graphics that can help to see how much damage the player can do with different values of their attacks, how much damage can he take from particular enemies, how much of the enemy attacks the average player can dodge (using the future playtesting sessions). This section will be hugely updated for the final release, because it can determine if the prototype is a viable product or not.

5.5.1 First balancing

The starting values of the enemies and the player were not calculated and were a complete disaster: it was almost impossible to beat the first boss.

To create this first balancing, there was hours of work dedicated on testing and adjusting them.

In order to solve this some data from the game was collected. A reference point was taken (in this case the player HP) and from there every number was build based on that.

5.5.1.1 Player stats

Based from the starting HP of the player, the rest of his stats were created:

Base Attributes	Amount
HP	1500
Attack	135
Critic Chance	0%
Critic Multiplier	1.5
Physical Defense	42
Magical Defense	21
Dodge	5.00%

Table 25: Player initial stats (first balance; self-made)

- Attack: 9% of the HP
- Critic chance starts at 0%
- Critic hits are multiplied by 1.5
- Physic defense: 2.8% of the HP
- Magic defense: 1.4% of the HP
- Dodge: 5%

The critic chance and dodge chance were decided to be low at the start, so they do not affect too much on the calculated and expected results.

The upgrades from each level on each stat is a 12% of their starting value, except critic chance and dodge which are 3% and 2% respectively:

Scaling Attributes	Amount
HP	180
Attack	17
Critic Chance	3.00%
Critic Multiplier	0.18
Physical Defense	6
Magical Defense	3
Dodge	2.00%

Table 26: Player scaling stats first balance (self-made)

Player skills damage were designed that each skill deals more damage and has more cooldown, the idea of this was so the player understands they can make the difference, as they can deal more damage than his regular attacks:

Attack	Damage %	Real Damage	Cooldown (s)	DPS	Type
Normal attack	100%	135	1	136	Physic ▾
Dagger throw	175.00%	297	5	60	Physic ▾
Dark shadows	325%	540	8	68	Magic ▾
TOTAL	600.00%	972	-	264	

Table 27: Player skills damage (first balance; self-made)

5.5.1.2 Store items

To start balancing the items, it is needed to know how many items can the player buy and in which point. For this, there is only needed to know the amount of coins each level has. The first level has only 11 coins, not so many, but enough to buy one normal item or even two cheap ones. On the second level, there are 60 coins, so if the player gets all the coins, the relevance of the items is huge at this point.

Each item gives different stats based on the item as the design explained, but these stats are based on the more expensive they are the better the stats/coin they give, encouraging the players to save to get the better items instead of buying the cheapest ones.

ID	Items List	HP	Attack	Crit. Chance	Crit. Mult	Physic Def	Magic Def	Dodge	Speed	Price
1	Axe of Destruction	0	100	10.00%	25.00%	0	0	0.00%	0.00%	25
2	Big Mace	0	80	0.00%	0.00%	0	0	0.00%	0.00%	12
3	Fire Dagger	0	40	10.00%	0.00%	0	0	0.00%	0.00%	8
4	Iron Helmet	500	0	0.00%	0.00%	35	0	0.00%	0.00%	6
5	Magic Helmet	500	0	0.00%	0.00%	0	35	0.00%	0.00%	6
6	Iron Armour	1000	0	0.00%	0.00%	75	0	0.00%	0.00%	20
7	Magic Armour	1000	0	0.00%	0.00%	0	50	0.00%	0.00%	15
8	Regular Belt	150	0	0.00%	0.00%	5	0	0.00%	0.00%	3
9	Iron Guantlets	250	25	0.00%	0.00%	10	0	0.00%	0.00%	9
10	Iron Boots	0	0	0.00%	0.00%	15	0	0.00%	10.00%	8
11	Runner Boots	0	0	0.00%	0.00%	0	0	0.00%	25.00%	8
12	Lucky Necklace	0	0	0.00%	0.00%	0	0	10.00%	0.00%	10

Table 28: Items list (first balance; self-made)

5.5.1.3 Player experience

Experience is another key point that will determine how stronger the player can get at each point. Each enemy gives experience when they die, and it needed to know at which points of the map the player is at each level, and of course, is stronger using his skill points.

Maximum, average and minimum player experience

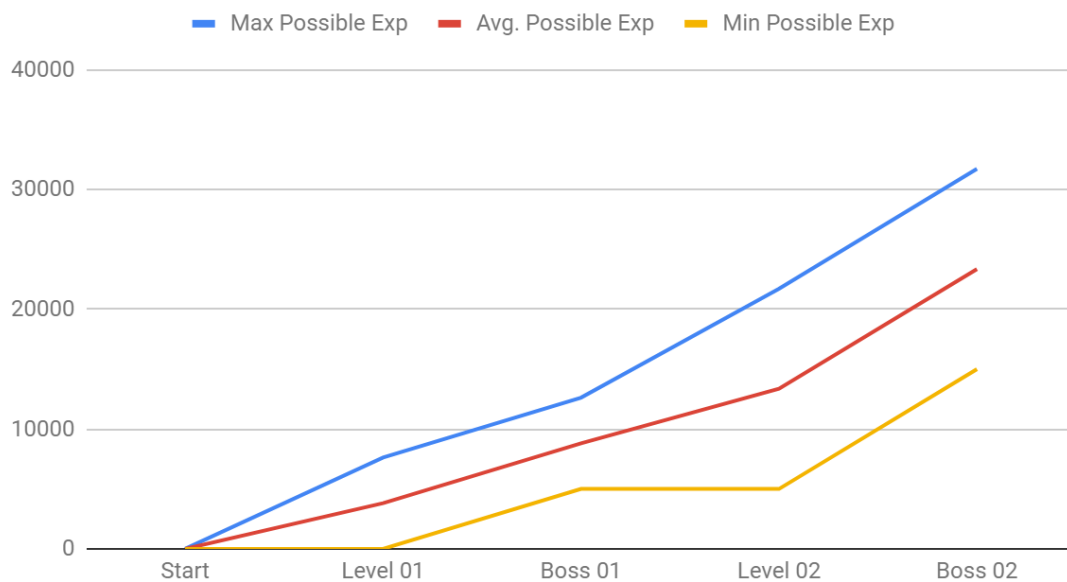


Figure 86: Maximum, average and minimum player experience (first balance; self-made)

5.5.1.4 Player potential damage and defense

Using all the previous data recollected, a graph with the DPS (Damage per Second) and the explosive damage from the player can be made in each level.

Player potential DPS and explosive damage

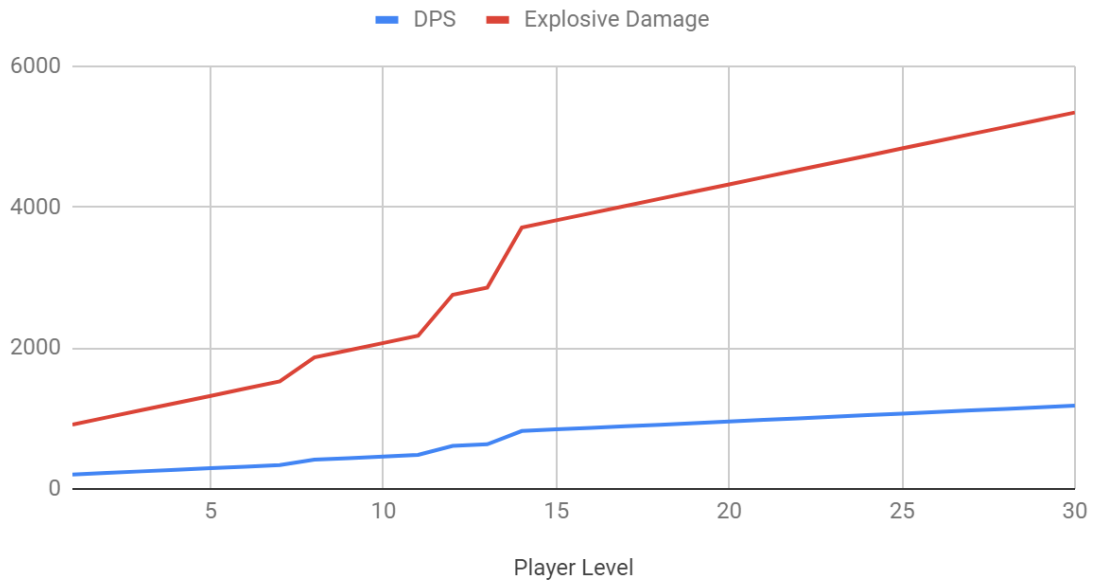


Figure 87: Potential player DPS and explosive damage (first balance; self-made)

The same way, it can be done with the physic and magic defense, to make sure the player always can be damaged.

Player potential physic and magic defense



Figure 88: Potential player physic and magic defense (first balance; self-made)

5.5.1.5 Enemies balance

Using all this data the enemies had been created to be challenging but not impossible to beat.

Here is the balancing done on the first level:

Level 01

List of Enemies	Number
Blue Hedgehog	5
Dwarf	1
Archer	1

Total Level Possible Exp
7625

Total Coins
11

Blue Hedgehog

Attributes	Amount
HP	325
Attack	145
Critic Chance	10%
Critic Multiplier	1.5
Physical Defense	25
Magical Defense	30
Dodge	5.00%
Exp Given	900

Skills	Damage	Cooldown	DPS	Type
OnTouch	150	-	-	Physic
ThrowSpikes	870	4	217.5	Physic

Dwarf

Attributes	Amount
HP	1400
Attack	125
Critic Chance	20%
Critic Multiplier	1.5
Physical Defense	40
Magical Defense	5
Dodge	2.50%
Exp Given	1425

Skills	Damage	Cooldown	DPS	Type
NormalAttack	125	1	125	Physic

Archer

Attributes	Amount
HP	650
Attack	275
Critic Chance	40%
Critic Multiplier	1.5
Physical Defense	10
Magical Defense	0
Dodge	10.00%
Exp Given	1700

Skills	Damage	Cooldown	DPS	Type
NormalAttack	275	1	275	Physic

Table 29: Level 1 enemies' stats (first balance; self-made)

On the first level, Ice lances that are shoot inflicted 150 true damage to the player and fire balls inflicted 100.

And here on the second level:

Level 02

List of Enemies	Number
Blue Hedgehog	3
Little Yeti	1
Archer	2
Fluffy Monster	3

Total Level Possible Exp
9100

Total Coins
60

Blue Hedgehog

Attributes	Amount
HP	325
Attack	135
Critic Chance	10%
Critic Multiplier	1.5
Physical Defense	25
Magical Defense	30
Dodge	5.00%
Exp Given	900

Skills	Damage	Cooldown	DPS	Type
OnTouch	150	-	-	Physic
ThrowSpikes	810	4	202.5	Physic

Little Yeti

Attributes	Amount
HP	2250
Attack	150
Critic Chance	0%
Critic Multiplier	-
Physical Defense	50
Magical Defense	50
Dodge	2.50%
Exp Given	1500

Skills	Damage	Cooldown	DPS	Type
NormalAttack	150	1	150	Physic
JumpAttack	175	1	175	Physic

Archer

Attributes	Amount
HP	650
Attack	275
Critic Chance	40%
Critic Multiplier	1.5
Physical Defense	10
Magical Defense	0
Dodge	10.00%
Exp Given	1700

Skills	Damage	Cooldown	DPS	Type
NormalAttack	275	1	275	Physic

Table 30: Level 2 enemies' stats (first balance; part 1; self-made)

Fluffy Monster

Attributes	Amount
HP	200
Attack	500
Critic Chance	0%
Critic Multiplier	-
Physical Defense	10
Magical Defense	40
Dodge	10.00%
Exp Given	500

Skills	Damage	Cooldown	DPS	Type
Explosion	312.5	0	-	Magic

Table 31: Level 2 enemies' stats (first balance; part 2; self-made)

On the second level, Ice lances that are shoot inflicted 175 true damage to the player, fire balls inflicted 125 and spikes, which did not appear on the first level, inflicted 225.

5.5.1.5.1 Bosses balance

Then the bosses needed to be more challenging and, specially, they cannot be beat in 10 seconds like some regular enemies. Because of that they have a lot more of health.

Monk (first boss):

Attributes	Amount
HP	5000
Attack	150
Critic Chance	5%
Critic Multiplier	1.5
Physical Defense	54
Magical Defense	28
Dodge	2.50%
Exp Given	5000

Skills	Damage	Cooldown	DPS	Type
Teleport	-		-	None ▾

Table 32: Monk boss stats (self-made)

Attributes	Amount
HP	10000
Attack	200
Critic Chance	0%
Critic Multiplier	-
Physical Defense	80
Magical Defense	80
Dodge	0.00%
Exp Given	10000

Skills	Damage	Cooldown	DPS	Type
NormalAttack	200	1	200	Physic ▾
JumpAttack	250	5	50	Physic ▾

Table 33: Ice Gorilla boss stats (first balance; self-made)

5.5.2 Balancing after first external playtesting

As seen in the playtesting results, with an exception of some players, the others found the game pretty difficult. However, it could be seen that most of them did not use any skill points or bought any items from the store. For this reason, it was logical to think that they found the game hard because they did not have upgraded their stats as they could. A proof of that is that every one of them except one player beat the first boss, meaning that it is not that hard at the beginning (before the difficulty starts raising).

For this reason, the next balance did not include a lot of changes, the player had a bit more health, from 1500 to 2500 and a bit more of attack, from 135 to 160. The upgrading percentage of those attributes was still 12%, so they also were updated from 180 to 300 in the case of the HP and from 17 to 20 in the case of the attack.

The last balancing change on the existing levels was on the first boss, its health was reduced from 5000 to 4000 HP.

5.5.3 Third level balancing

The third level was created between sending the first playtesting release to testers and receiving all of their responses. Having this feedback helped to create a more balanced and fair stats for those enemies, some new, some already seen on the first two levels but with increased stats.

5.5.3.1 Enemies balance (3rd level)

This was the stats from the enemies created on the first level, which are obviously harder than the previous ones because so is the player in this point.

This level is also characterized by the number of enemies it has, which is much more than the previous ones.

Level 03

List of Enemies	Number
Dwarf	1
Lizard Monster	3
Blue Hedgehog	5
Archer	7
Fluffy Monster	2
Imp	3

Total Level Possible Exp
 28500

Total Coins	Health potions
32	4

Dwarf

Attributes	Amount
HP	2500
Attack	150
Critic Chance	20%
Critic Multiplier	1.5
Physical Defense	40
Magical Defense	5
Dodge	5.00%
Exp Given	1500

Skills	Damage	Cooldown	DPS	Type
NormalAttack	150	1	150	Physic

Lizard Monster

Attributes	Amount
HP	3000
Attack	250
Critic Chance	15%
Critic Multiplier	-
Physical Defense	100
Magical Defense	100
Dodge	0.00%
Exp Given	2000

Skills	Damage	Cooldown	DPS	Type
NormalAttack	250	1	250	Physic
Horn	250	1	250	Magic

Figure 89: Level 3 enemies' stats part 1 (self-made)

Blue Hedgehog		Attributes	Amount	Skills	Damage	Cooldown	DPS	Type
HP			325	OnTouch	180	-	-	Physic
Attack			175	ThrowSpikes	1050	4	262.5	Physic
Critic Chance			10%					
Critic Multiplier			1.5					
Physical Defense			25					
Magical Defense			30					
Dodge			5.00%					
Exp Given			1000					

Archer		Attributes	Amount	Skills	Damage	Cooldown	DPS	Type
HP			750	NormalAttack	350	1	350	Physic
Attack			350					
Critic Chance			40%					
Critic Multiplier			1.5					
Physical Defense			10					
Magical Defense			0					
Dodge			20.00%					
Exp Given			1750					

Figure 90: Level 3 enemies' stats part 2(self-made)

Fluffy Monster		Attributes	Amount	Skills	Damage	Cooldown	DPS	Type
HP			400	Explosion	500	0	-	Magic
Attack			800					
Critic Chance			0%					
Critic Multiplier			-					
Physical Defense			10					
Magical Defense			60					
Dodge			0.00%					
Exp Given			750					

Imp		Attributes	Amount	Skills	Damage	Cooldown	DPS	Type
HP			1250	NormalAttack	175	1	175	Physic
Attack			175					
Critic Chance			35%					
Critic Multiplier			1.5					
Physical Defense			20					
Magical Defense			20					
Dodge			10.00%					
Exp Given			1250					

Figure 91: Level 3 enemies' stats part 3(self-made)

On the third level, Ice lances that are shoot inflicted 200 true damage to the player and spikes inflicted 225. Fire balls did not appear on this level.

5.5.3.2 Boss balance (3rd level)

In the case of the third boss, the stats were the following:

DEMON

Attributes	Amount
HP	20000
Attack	250
Critic Chance	10%
Critic Multiplier	-
Physical Defense	50
Magical Defense	50
Dodge	0.00%
Exp Given	25000

Skills	Damage	Cooldown	DPS	Type
NormalAttack	250	1	250	Physic
Explosions	960	7	137.14	Magic

Summonings

Imp
Every 20 seconds

Attributes	Amount
HP	2250
Attack	150
Critic Chance	0%
Critic Multiplier	-
Physical Defense	50
Magical Defense	50
Dodge	2.50%
Exp Given	0

One Eyed Imp
Every 25 seconds

Attributes	Amount
HP	2250
Attack	150
Critic Chance	0%
Critic Multiplier	-
Physical Defense	50
Magical Defense	50
Dodge	2.50%
Exp Given	0

Figure 92: Demon boss stats (self-made)

5.5.4 Balancing after second external playtesting

In the end of the playtesting it was seen that the balance was pretty fine but in some parts was seen to be too easy.

Enemy	Level	Attribute	Old value	New value
Monk	Boss 1	HP	4000	5000
Archer	Level 2	HP	650	750
Archer	Level 3	HP	750	850
Hedgehog	Level 2	HP	325	375
Hedgehog	Level 3	HP	325	425
One Eyed Imp	Boss 3	Shoot cooldown	5	3
Demon	Boss 3	HP	20000	22500
Demon	Boss 3	Physical Defense	50	100
Demon	Boss 3	Magical Defense	50	100

Table 34: Balancing (post 2nd playtesting)

On the second level there was some other changes as well:

- Added two new enemies on the middle of the map, one archer and one little yeti.
- Increased the two last rocks with spikes speed from 2 to 6.
- Added a new rock with spikes that moves around the 3 last platforms before reaching the boss door.

And finally, on the third level, the last platform spikes that surround the boss key had their speed were reduced from 8 to 6.

In exchange for this little upgrade for the enemies, the player can receive a bit more experience on the second level. In the end of the 3rd boss he can be level 30.

5.5.5 Final balancing

With all the changes, the final stats of the player and every enemy in the game can be seen in the following link:

https://drive.google.com/open?id=1cptQJENx6Q3YfzNnTH5PVaAIzcEa8yKEj6Z_KaPx5PU

5.5.6 Polish

While the second external playtesting results were coming and until the end of the project, a lot of polish has been made to the game.

5.5.6.1 Demon boss and its level redesign

The third boss level was redesigned because it made no sense to have such platforms for jumping on them with the player when the boss was so big, those platforms only worked for slowing up the boss and the player got hit when he was up there anyway, so they were removed.

Also, the first approach of the boss was quite poor: all the bosses had their own special skills and the Demon could just summon other enemies, so there was its skill implemented were explosions occurred in random places on the map, and if the player was close to those points, he got hurt.

However, a lot of times those explosions felt they would never hit the player because the map was quite large, and it was a bit strange that an explosion occurred near the player. So the random position on all the map became a random position near the player in a close radius circle.

5.5.6.2 Tutorial level

After the first playtesting there was the idea of a tutorial level to explain the players all the mechanics and all the features of the game. It was started but never ended because it was calculated to cost way too much time to be done well and it was considered better to have the three existing levels done better than have them worse and have a probably not really good tutorial level.

5.5.6.3 Spawners sprite change

Ice lance and fire balls spawners were not understandable as they should be with the last particle effect change, and not only that, they could be confused with the teleporters, so after searching for some ideas of spawners two sprites were created to change them.



Figure 93: Final ice lance spawner (modified sprite)



Figure 94: Final fire ball spawner (modified sprite)

5.5.6.4 Typography change

Unity's default typography is Arial, it has a good legibility, but the esthetic does not really feels like a cartoon game like this one. For this reason, all the texts from the game changed to "Lucida Grande Bold", which is more cartoon like and also has a good legibility.



Figure 95: Arial (left) vs Lucida Grande Bold (right) comparison (self-made)

5.5.6.5 Elastic spring sprite

The elastic spring behavior was created on the first external playtesting and no one said it was not understandable because they only find out that the box with a star makes the player jump high, but to make it more accurate this sprite was replaced with an elastic spring that even has an animation when the player touches it, making it feel what it actually is.



Figure 96: Elastic spring sprite sheet (self-made, from bought sprite)

5.5.6.6 UI aesthetic redesign

The UI that was bought was in a much different color that the game had, so instead of using the bought assets, a new UI was created that had a similar tone color palette.

You can see the difference in the following comparison images:



Figure 97: Main menu (before)



Figure 98: Main menu (after)

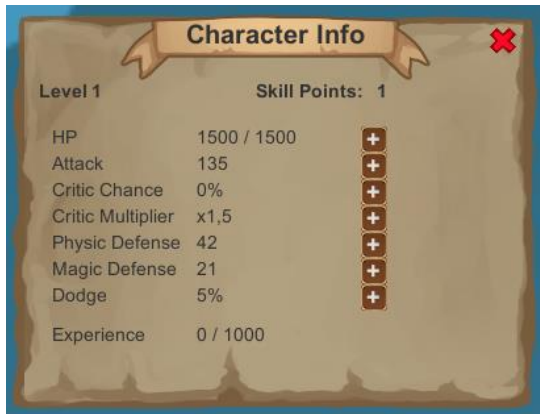


Figure 99: Character info (before)



Figure 100: Character info (after)



Figure 101: Store (before)



Figure 102: Store (after)



Figure 103: Settings (before)

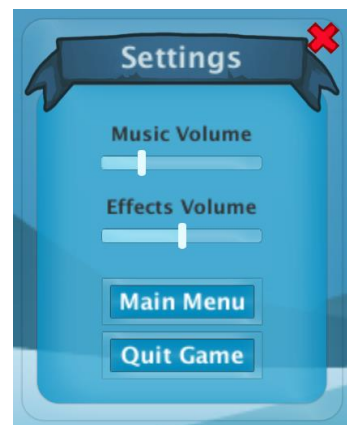


Figure 104: Settings (after)

5.5.6.7 Door animation

An animation was created when the player opens the door to start the new level or to enter the boss room. The audio of the opening door was already created, and it felt better with an animation.



Figure 105: Door animation (modified sprite)

5.5.6.8 Boss levels closing door

On the boss fights, it was needed to close the entrance because otherwise, the second and third bosses were too big to fit in there and the player could kill them by simply throwing knives patiently. To avoid that a door that closes the entrance when the player steps in and the boss fight starts was created.



Figure 106: Boss closing door (modified sprite)

5.5.6.9 Level 3 new platform

On the third level there was a little level design problem that in the middle of the map there was no way to turn back to the level, so a little platform was created so the player can jump back and make his way back.

5.5.6.10 Main menu music

All the game had music and sound effects, but the menu was silent and that looked a bit strange, so a music was added to it. The music was extracted from this webpage: <https://opengameart.org/content/the-kings-crowning>.

5.5.6.11 Victory screen

After killing the third and last boss, nothing happened in the game, so when that happened and the last door was opened a new screen was created indicating to the player that he completed the prototype.



Figure 107: Victory screen (self-made)

5.5.6.12 Main menu introduction

An image when the player opens the game has been created that fades out leading to the main menu when the music starts was also implemented.



Figure 108: Main menu cover (self-made)

5.5.6.13 Bugfixes

With the development of the project a lot of bugs were also seen and fixed. In the following link there is the list of all 120 bugs found during the project development:

https://drive.google.com/open?id=1dDU6slrvOStm2y0NaY1YauRoj12V_OU-kxGDkV04uyU

Almost all of them were solved, except for some that were not really important and it was considered there were other more important features to develop or other problems to solve than those.

5.6 Website

In the end of the project, it was decided to create a little website for it. This was not planned at all, but the final result was considered a success and with the little time there was left, this website was created:

<https://powerchase9.webnode.es/>

The website was created using the free mode of “webnode”. It is a website that provides you with the server and different templates to use. The only part that would have been used if the paid option was bought would be the direct download button (clicking a button and instantly download the game files), that there is a watermark in the end of the website page that says it was created using webnode and that you can choose the exact URL, therefore the 9 in the existing URL.

The first point was simply solved by creating a link button that directs the user to a download link where he or she can download the game, and the second and third points were quite irrelevant in the prototype.

The website will be finished the 14th of July 2019.

6. Conclusions and future work

In conclusion, the prototype of the game was a success, both the general and all of the specific objectives were accomplished, and it can be considered more than a prototype.

6.1 Project times

In the end of the project, the timing was a bit larger than the expected. Nevertheless, this is a really common thing in video games, and not only that, a lot of new things that were not planned at the beginning were created and improved the game. That is the reason why the planned hours did not match exactly, but it is a pretty accurate timing of what I expected.

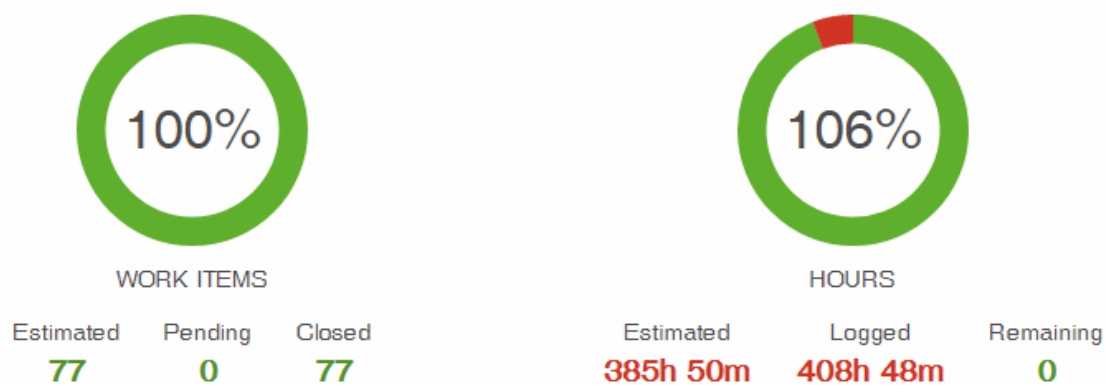


Figure 109: Estimated and real hours

From all the real time spend on the project development (408 hours and 48 minutes), the tasks breakdown is the following one:

- Programming: 165 h 10 m
- Writing the memory: 81 h
- Design: 58 h 45 m
- Bugs: 54 h 10 m
- Research: 26 h
- Testing: 12 h
- Management: 7 h 43 m
- Marketing: 4 h

6.2 Planned objectives

6.2.1 General objective

The general objective of creating a prototype was more than accomplished, if you look a prototype of a videogame and you compare it to this one, this seems more a final video game than a prototype. Of course, it is not a completed video game because it lacks some things that

will be seen in “6.3 Future work” and even more levels because it can be beat in a short time; but the game features and details that were created are not at all a prototype.

Because of those reasons, the general objective was a huge success.

6.2.2 Specific objectives

6.2.2.1 *Designing the whole game*

Clearly the design was created and finished, if not the prototype could not have been made. Even the little story was created when it was not an objective, but it is always a plus for a game prototype and it could be continued and become more interesting in the future.

6.2.2.2 *Manage all the tasks satisfactorily*

All the initial tasks that were raised were accomplished. Of course on the development of the prototype there were new created tasks to polish and make the game better, and that was taken into account at the beginning because it was obvious that after the playtesting there will be bugs and new possible features to develop, that’s exactly the intention of the agile methodology chosen, because it is the only way to create a fun game.

Those new tasks created an overdue to the initial planned hours reflected on “6.1 Project times”, however this was all calculated and the prototype could have been done and perfectly be a fun game without most of them, but the polish and details feeling would not have been the same, reason why it was decided to put an extra effort at the end of the project to achieve this tasks.

6.2.2.3 *Achieve a fun gameplay*

That is probably the most difficult objective to check if it was accomplished or not, because it is the most subjective thing and it can be amazing for someone and boring for someone else.

However, comparing the second and last external playtestings results we can see that at least the people that played the game had fun with it and ranked the prototype with a good average score of 7.1 out of 10, and even after this more polish to the game had been created, so it can be considered a fun game prototype.

6.2.2.4 *Balance the game*

This point is also a bit difficult to know if it is accomplished or not so much because it is a subjective thing that depends on the skills of every player. However, it can be considered accomplished as well because of the difference between the first and the second external playtesting, where the players opinions were from too difficult to a balanced difficulty between those playtestings.

6.2.2.5 *Implement a good UI system*

The starting designed UI was created and implemented at its fully and there is a variety of options in the UI. It has the store, the character information panel, the settings, the main

menu, an ending menu, all the HUD for the player and the popups both from the skills description and the damage the characters receive.

For this variety we can see that the UI system is considered as good, especially for a prototype.

6.2.2.6 Carry out a useful playtesting

As said in the playtestings explanations, they were astonishingly useful; there is no doubt that without those playtestings the final prototype would have been way worse than it actually is.

6.3 Future work

It is a game prototype so, of course there is a lot of future work to do in order to complete the full game to be ready for the marketplace.

Nevertheless, the prototype itself is really complete and went beyond expected, meaning that most of the workload goes into expanding what is already created, things such as giving the player more skills, more and different enemies and bosses and more levels to complete.

Some of the things that are not simply expanding what already exists and that after the playtestings and experience on the project would made in order to complete an ending product game would be the following:

6.3.1 Player skills

Creating a system for the player to obtain skills instead of having them all from the start like in this prototype. This was not applied to the prototype because there were few skills and there is no skill tree like other games have so the player can choose which skill he wants from different choices.

Those skills could be obtained simply each 5 player levels or each time the player defeats a boss. One this happens the player obtains a point that can be exchanged for unlocking the skill that the player chooses.

One idea of a skill that was though was to turn enemies to allies for a short time.

6.3.2 Crouching

Crouching could be a good mechanic in order to avoid ranged attacks from both enemies or ice lances, it can improve the control the player has and feels in the game. This was not possible to do in this prototype because there was no animation in the bought assets of any character that can stay crouched and creating an animation for only this feature would have cost way much time.

6.3.3 Buy health potions

The option of buying health potions from the store can be a good improvement because in order to defeat some bosses they can be really useful, as the combats last longer.

6.3.4 List of bought items or inventory

Create an inventory or a list of the items you have bought so the player can see the items they possess.

6.3.5 Level score and achievements

A score in the end of each level or in the end of each chapter if the game is expanded and the main story is divided by chapters that narrate the story and also achievements to complete such as collecting all the coins on the game or defeating a number of enemies that gives rewards to the player.

Those features will highly increase the replayability of the game.

6.3.6 Save and load

A save and load and managing files is a must in order to create the full videogame, in the case of this project it was never a priority because there are only 3 levels, but if it scales and in the end the game has, for example 10 levels, the player cannot start from the beginning each time because he will become bored easily.

6.3.7 Difficulty levels

Creating at least two different difficulty levels. Most of the players found the prototype balanced in the last playtesting; however, there are players that enjoyed a lot more the platforming and the player mobility than the combat and they liked more to play with the player mobility. A lower difficulty could be implemented to reach those players.

7. Webography

- HumbleBundle, website for buying assets. www.humblebundle.com. Consulted 5 November 2018.
- Guacamelee! 2. <http://www.drinkboxstudios.com/press/sheet.php?p=guacamelee!2>. Consulted 15 February 2019.
- DrinkBox Studios. <http://drinkboxstudios.com/blog/about-us/>. Consulted 15 February 2019.
- Guacamelee! 2 developers' interview. <https://gamingbolt.com/guacamelee-2-interview-coming-home-to-roost>. Consulted 15 February 2019.
- Salt and Sanctuary post. https://www.reddit.com/r/saltandsanctuary/comments/4c9955/how_was_the_game_made/. Consulted 15 February 2019.
- Ska Studios. <https://ska-studios.com/team/>. Consulted 15 February 2019.
- Salt and Sanctuary development. https://www.reddit.com/r/saltandsanctuary/comments/4as6qq/what_do_you_think_the_budget_for_salt_and/. Consulted 15 February 2019.
- Game Wuppo. <http://wuppogame.com/about/>. Consulted 15 February 2019.
- Knuist and Perzik (Wuppo). <http://knuistperzik.com/>. Consulted 15 February 2019.
- Wuppo development blog. <http://wuppogame.com/news/developmentblog2015/>. Consulted 15 February 2019.
- Warlocks 2: God Slayers. <https://www.indiedb.com/games/warlocks-2-god-slayers>. Consulted 15, February 2019.
- Warlocks 2: God Slayers development article. <https://www.moddb.com/games/warlocks-2-god-slayers/features/warlocks-2-gods-slayers-how-frozen-district-navigates-the-indie-landscape>. Consulted 16 February 2019.

- Treasure Adventure World press kit information. <http://robitgames.com/press/sheet.php?p=Treasure%20Adventure%20World>. Consulted 16 February 2019.
- Treasure Adventure World article. <http://treasureadventureworld.com/2018/04/a-look-at-taws-launch-and-future/>. Consulted 16 February 2019.
- Timespinner Kickstarter. <https://www.kickstarter.com/projects/lunarraygames/timespinner>. Consulted 16 February 2019.
- Dead Cells engine used. <https://steamcommunity.com/app/588650/discussions/0/2741975115066308346/>. Consulted 16 February 2019.
- Motion Twin information (Dead Cells). <https://motion-twin.com/presskit/81#credits>. Consulted 16 February 2019.
- Dead Cells development post. <https://www.mcvuk.com/development/when-we-made-dead-cells>. Consulted 16 February 2019.
- Catmaze game information. <https://www.moddb.com/games/catmaze/news/dev-diary5>. Consulted 17 February 2019.
- Catmaze developer twitter. <https://twitter.com/rbspade?lang=en>. Consulted 17 February 2019.
- Game unworthy. <http://www.unworthygame.com/Press/index.php>. Consulted 17 February 2019.
- Unworthy game information. <https://www.moddb.com/games/unworthy>. Consulted 17 February 2019.
- Unworthy interview with Aleks Kuzmanovic. <https://stephansuniverse.wordpress.com/2018/02/02/interview-with-unworthy-designer-aleks-kuzmanovic/>. Consulted 17 February 2019.
- White book of Spain video game development 2018. <http://www.dev.org.es/images/stories/docs/Libro%20Blanco%20DEV%202018.pdf>. Consulted 13, March 2019.
- Newzoo global games forecast. <https://newzoo.com/insights/articles/newzoo-cuts-global-games-forecast-for-2018-to-134-9-billion/>. Consulted 13, March 2019.
- Newzoo top revenue companies 2017 and 2018. <https://newzoo.com/insights/rankings/top-25-companies-game-revenues/>. Consulted 13, March 2019.

- Unity basic features. <https://unity3d.com/unity>. Consulted 13, March 2019.
- Unity editor features. <https://unity3d.com/unity/editor>. Consulted 13, March 2019.
- Unity 2018.3 release detailed info. <https://blogs.unity3d.com/2018/12/13/introducing-unity-2018-3/>. Consulted 13, March 2019.
- Unity 2018.3 release info. <http://www.cgchannel.com/2018/12/unity-technologies-releases-unity-2018-3-in-beta/>. Consulted 13, March 2019.
- Unreal engine 4.21 release notes. <https://www.unrealengine.com/en-US/release-notes>. Consulted 13, March 2019.
- Unreal engine information. <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>. Consulted 13, March 2019.
- Unity and Unreal comparison. <https://www.quora.com/What-are-the-differences-between-Unity-Engine-and-Unreal-Engine>. Consulted 13, March 2019.
- Steam store, with reviews and scores from games. <https://store.steampowered.com/>. Consulted 13, March 2019.
- Metacritic. <https://www.metacritic.com/game>. Consulted 13, March 2019.
- IGN. <https://es.ign.com/pc/>. Consulted 13, March 2019.
- PEGI ratings. <https://pegi.info/what-do-the-labels-mean>. Consulted 14, March 2019.
- The Design and math of incremental games. <https://gamedevelopment.tutsplus.com/articles/numbers-getting-bigger-the-design-and-math-of-incremental-games--cms-24023>. Consulted 18, March 2019.
- Free cartoon particle effects. <https://assetstore.unity.com/packages/vfx/particles/cartoon-fx-free-109565>. Consulted 24, May 2019.
- Open source particle effects. <https://codemanu.itch.io/>. Consulted 23, May 2019.
- Unity issue with lighting. <https://issuetracker.unity3d.com/issues/regression-serializedobject-target-has-been-destroyed-appears-after-environment-lighting-change>. Consulted 22, June 2019.
- Unity documentation. <https://docs.unity3d.com/Manual/index.html>. Consulted multiple times on the project development.