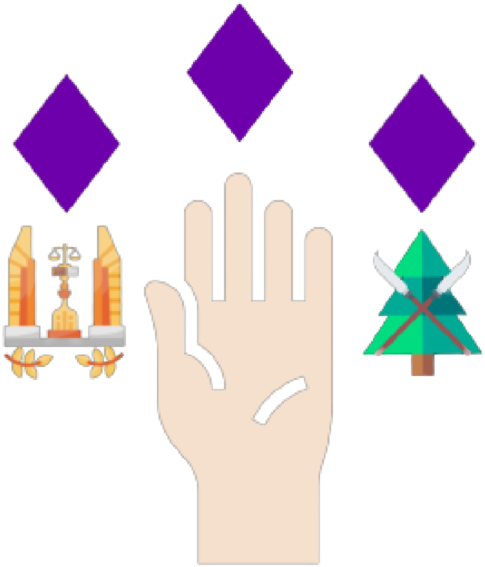
Multiverse of Geekiness

A S2 Project

April 17, 2023



Pierre Chain - Alexandre Courrieu - Damien Duthou - Arthur Maronian

Contents

1 Introduction

Multiverse of Geekiness is meant to be a single/multi -player adventure game that will rely on a constantly evolving gameplay. Such a goal will be achieved thanks to different geek-culture “universes” that the player(s) will have to traverse and survive to.
We chose such a format because it can firstly give us a lot of flexibility over how to implement new mechanics between the different zones. The format will also allow the game to keep the player(s) interested not only thanks to varying very different backdrops and challenges, but also as it will conserve a clear linearity. This linearity will ensure that the player(s) keep(s) enjoying the adventure aspect and, if playing in multiplayer mode, the PvP[1] feature that will aim to be an immersive experience thanks to the race-like gameplay.

1.1 Information Sheet

Project Type: Video Game
Genre: Adventure
Game Style : Exploration/ Combat (PvP and PvE[2])/ Race/Mission
Platform: PC Windows
Engine: Unity
Number of Players: 1 or 2

1.2 Case Study

;Multiverse of Geekiness, will aim to be a trip among some aspects of the internet culture as much as it is meant to be a game. Truly unlike other action packed video games, MoG’s[3] objective will be to provide a relaxing atmosphere in single player, but also in multiplayer where the PvP even if allowed will stay occasional due to the maze-like arrangement of the dimensions. Therefore, the game is meant to be unique thanks to its gameplay rather than its scenario or even, final goal.
This project is first of all a great self-assessment opportunity, we all are really interested in computer-science and are willing to see how good we can perform in our first large-scale project. Second such a project will also be a great learning experience, in terms of the actual tech-skills we are going to develop but, also more importantly in terms of group working coordination.

[1] *Player versus Player, when two players fight with each other.*
[2] *Player versus Entity, when a player fights a creature from the game.*
[3] *Multiverse of Geekiness, the preferred abbreviation to refer to the game.*

2 Game description

Our Hero, whose name is yet to be decided, has made a terrible discovery: his world is about to die. luckily for him, he found ancient murals describing how an ancient civilization was able to save their planet using a rare stone. However, the location of said gem is not disclosed, only the tale of it's guardian.

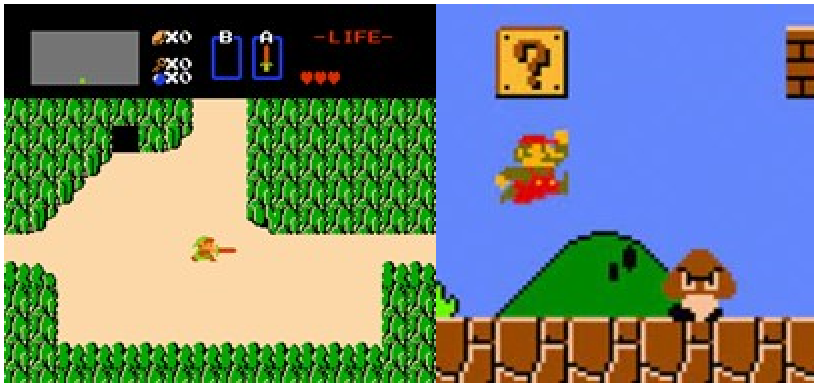
2.1 Gameplay

2.1.1 Goal

As explained before, the player will have to find a gem to save his world. To do so, he will have to access the End Dimension as quickly as possible. He will then have to defeat the guardian and search for the stone, that can only be spotted once the guardian has be defeated.

2.1.2 Multiverse

To find the End Dimension, the player will have to Jump through Portals and travel through the multiverse. Each World / Dimension / Universe will represent different game types and use mainly two views, namely from above and the side. Most of the mechanics will stay the same throughout the entire game, but some of them will be specific to the universe where the player is currently. Using different views and game styles should be the main priority of the project, as having smooth transition between styles is essential for the game experience.



Example of a view from above and a side view (in that order)

Dimension design	**Common games of that design**
2D top-down scrolling shooter	1942,
Platformer	Metroid Vania, Mario
Large map adventure	Zelda, Pokemon

Non-exhaustive list of possible dimensions

2.1.3 Entities

The game will feature different entities, or mobs, that can hurt the player or help him. The player(s) can then utilize the different features to their advantages, that is using peaceful entities for themselves, or using aggressive entities to block or slow down the other player.

2.1.4 Fight mechanics

One of the key feature of the game is the use of different game styles. This should allow for a lot of variations in fight mechanics and weapon use. Each game style has indeed preferred fight mechanics. A side view will make the most use of barehand attacks and complex movements, as seen in the Street Fighter franchise. A top-down scroller will be best implemented with long range weapons, while the above view will favor melee weapons.
Those fight mechanics can of course be excluded from certain dimension, where the player won't fight entities but rather have to deal with the terrain, much like a parkour game.

2.2 Beginning

2.2.1 Solo Mode

In the solo mode, the player will spawn in a hub with basic loot. He will then have a choice between several Portals, each one allowing the player to go to a different world. The player will be able to return to his spawn hub if he finds the necessary Portals. Those Portals might be different from the one he took to go to one of the world.

2.2.2 Multiplayer Mode

The multiplayer mode will follow the same mechanics. The only difference will be that each player will be able to access his, and only his spawn hub, whether it is from dying or by following a portal.

2.3 End fight

The end fight should be the last and most exciting part of the game. The Boss, whose form is still to be decided, will guard the gem. While still alive, the boss will prevent the player(s) from seeing it. The preferred game mode will then be an above view, allowing much more freedom of movement to player(s) and entities.

3 State of the Art

This game will be a combination of multiple game styles, all of them in 2D. This choice is both artistic, because the 2D art style is more interesting and fun to play, and practical because it is easier to draw and 2D projects tend to look better than Blender Models when made by non-professionals. The main art style/game styles chosen being:

1. Adventure games such as the first “The legend Of Zelda” game with view from up top and ground movement only.
2. Metroidvanias such as Super Metroid or Castle Vania with side view with only forward and upward movement.

Other game styles will come into play as the development goes on.

3.1 Inspiration

On the art style, gameplay and the lore of the game, inspiration will be taken from mainly geek culture because of its importance and since most project members are very fond of it.

The lore will be a lot like Ready Player One, mashing multiple universes into one, giving the players a unique experience meeting different game styles along the way. The goal being here to break traditional game codes in which only one style of gameplay is encountered throughout the game. This has already been done before and we intend to make it as enjoyable and ambitious as the previous examples of such diversity.

Graphic and musical art style will be retrogame-like using pixels and electronic music. This, both in attempt to pay homage to these games, that have planted the foundations of today’s game design, and to have an enjoyable art style that doesn’t require too much expertise to be made. Also, from a practical point of view, pixels and electronic music are easier to make for non-professional artists and easier to find resources for online.

Finally, the game will have this race and path-crossing themed gameplay, making players in some sort of rivalry for who can complete the game the fastest and measuring different skills (adventure, platforming, PvE, PvP, Strategy,...). We want to make players both racing independently but also being in direct competition regularly. The idea is to give an advantage to the player that arrives first at the competition and a go-ahead to the one that wins said competition.

4 Technical Consideration

4.1 Game

4.1.1 Multiplayer

As the game is rather short, doing several game one after another might lead to the boredom of the player. To introduce some diversity in games, the game will allow for a multiplayer mode. The multiplayer part of the game will be built using Netcode Framework which is a popular Unity library that simplifies many complex implementations.

4.1.1.1 PvP

Since two players can be connected to the same game and since they pursue the same goal, which is reaching the End Dimension to find the gem, it will be interesting to implement PvP. This means that the players will be able to fight each other and kill each other, in order to get the loot from the other player, or simply slow down the other player's progression.This implies calculating the latency in order to not disturb players' experience.

4.1.2 Artificial intelligence

The Game will feature many entities including a few bosses that serve as milestones for the player's progression.
In order to implement the Artificial Intelligence correctly, the chosen method is an Ad Hoc implementation where each action will be mapped to a set of specific rules.

4.1.3 Path Finding

As players can move on different axis, the entities should move accordingly. With the perspective of an enemy attempting to attack the player, this specific entity that is within a specified range from the player hitbox will use a predefined Dijkstra algorithm to follow the player. Non-hostile entities might flee if the player gets too close, or simply have no interaction whatsoever with the player. Some terrain might prevent entities from moving, following the player, or simply prevent the entity from walking or jumping on this terrain (gaps or terrain that do damage for example).

4.1.4 Ad Hoc method

The game's Ad Hoc method consists of a table where each action gets mapped to a specific defined rule. A rather simplistic implementation will be shown in this document. The purpose of this method is to ensure that every action that an Entity can do is strictly defined by the game's developer. This approach also allows more ”real” actions rather than random ones

4.1.5 Ruleset

RULES	ACTIONS
Player within defined range	Hostile entities follow the player accordingly
Player within defined range	Passive entities flee the player accordingly
Map in top-view	Player moves from left to right and front and back, but cannot jump
Map in top-down scroller	Player moves from front to back, and can jump
Item within the Entity's range	Entity will try to pickup the item
Entity's inventory full	Item will not be picked up
A gap is found on the path between the Entity and its destination	Entities will ”jump” over the gap or turn around
Player enters a Portal	Player spawns in a differnt dimension

4.1.6 Generating entities

Some parts of the game will require randomly generated entities. Two approaches have been considered in order to fulfill that purpose. The first one is a classic random generation by number picking. The second one provides a more interesting perspective. Instead of picking random numbers we will apply a coefficient that is related to the plater item.

4.1.7 Game visual

4.1.7.1 General Theme

Since the game is set in a rather futuristic world, the graphics should mainly be retro-futuristic. As the point of the game is to switch between game modes, it would be also interesting to switch between styles.

4.1.7.2 Portal

The player is going to have to travel across dimensions to finish the game. We decided that all portals, except the one leading to spawn, should have the same appearance. This way, the player will have to explore the world, and each game will be different from the one the player did before, much like random structure generation in Minecraft.

4.1.7.3 Views

There is several possibilities for handling the view system. One of them would be to create a 3D game, using cubic textures, and utilizing the camera feature of Unity. Of course, this will affect the performance of the game in a negative way.
The other method includes generating different scenes, but might be harder to implement in the game.
One of our main tasks is to test both method and decide which solution will be easier to implement, while allowing fast gameplay.

4.1.8 Audio

As none of us, ever worked on developing audio to be used in games, this part will be tricky. We will mainly try to use free sound library to reduce at maximum the cost of developing the game. Nonetheless, we might, at some point, have to record or create our own sounds, adding another challenge to this project.

4.1.8.1 Music

As previously stated, game music will mainly be electronic. Since the game is mainly a race, the music should produce an effect of adrenaline for the player. If implemented, the Boss Fight music should be quite different from the rest of the game's music, and should correspond to a rather epic style. One of the drawbacks of having music when fighting the Boss is that it may come out as a distraction for the player, which is why we're not sure to implement it.

4.1.8.2 Sound effects

Some actions will feature attributed sound effects, also in a retro style. For example a player attack should produce sound to keep the game dynamic. Entering and leaving Portals will also produce sound. Some entities might also feature their own sound effects.

4.1.9 In-game experience

Using only 2D and rather simplistic graphics should allow a completely smooth gameplay with minimal lag, even on lower end computers. It is also a way to attract players through gameplay and storytelling rather than by using astonishing visual effects.

4.1.10 Testing

During the project and after it's completion, we'll have to thoroughly test in order to detect any bugs or unwanted mechanics. This'll be achieved by the team throughout the project, but also at the end of it. Once the game has passed all of our requirements, it will be shared to a bigger group of people, such as family and friends. Their experience playing the game will allow us to make final improvements.

4.2 Task Division

4.2.1 The Team

4.2.1.1 Pierre Chain

Pierre's extensive knowledge of game universes naturally places him as the storyline and structure manager of the game. He will ensure that the game stays coherent.

4.2.1.2 Alexandre Courrieu

Alexandre will assume the role of the spokesperson. He'll be in gharge of coordinating the presentations and the documents. He'll also be in charge of the website, as he is the one with most experience in this domain of the group.

4.2.1.3 Damien Duthou

Damien is currently considered as our artistic director. His sharp opinion on art styles and game designs will be immensely valuable to judge the beauty of the game. He will not work on the graphics creation per say.

4.2.1.4 Arthur Maronian

Arthur's leadership made him the top pick for the role of group leader. His sense of humor, paired with his seriousness and tirelessness will ensure a serious but fun atmosphere and the group, as well as preventing conflicts.

4.2.2 Task Distribution

Legend: X – main manager
O – secondary manager

Tasks/Member	Pierre Chain	Alexandre Courrieu	Damien Duthou	Arthur Maronian
Multiplayer		X		O
AI	O		X	
Map Graphics			X	O
Entities Sprites	O	X		
Backgrounds	O	X		
Level design	X			O
Game design	X		O	
Class ”tree”		X		O
Audio		O	X	
Website	O	X		
Storyline	O			X
Items	X		O	
Ruleset			X	O
Player abilities			O	X
Entities abilities		O		X

4.2.3 Task Completion

Tasks/Date	First presentation	Second presentation	Final presentation
	(6 - 10 March, 2023)	(17 - 21 April, 2023)	(29 May - 9 June, 2022)
Multiplayer	30%	70%	100%
AI	20%	65%	100%
Map Graphics	20%	30%	100%
Entities Sprites	10%	20%	100%
Backgrounds	10%	20%	100%
Level design	20%	50%	100%
Game design	50%	85%	100%
Class “tree”	80%	100%	-
Audio	0%	50%	100%
Website	80%	90%	100%
Storyline	70%	100%	-
Items	30%	50%	100%
Ruleset	50%	85%	100%
Player abilities	50%	85%	100%
Entities abilities	50%	85%	100%

4.3 Publication

4.3.1 Website

The associated website will have two main goals. The first one will be communication, by presenting the game and our progress. The second part should feature more technical details about the game. Among those details, the code source should be available to the public, as well as a “Wiki”. It’ll also feature the class tree and the ruleset used in the game. It will be redacted using HTML5, CSS and JavaScript and will be hosted and referenced by Netlify. We’ll then have to research SEO in order for our website to be easily accessible to the public.

4.3.2 Documents

For the project to be conducted smoothly, will need to redact several documents, ensuring that all team members know the state of the project and agree on game mechanics. Those documents include:

1. Class tree, allowing us to have a clear overview of how different mechanics are linked and what needs to be coded.

2. Ruleset will be a reference about what can and cannot be achieved by the avatar, in order to avoid conflicts while building the game

3. Planning, to keep track of our progression, have clear deadlines and be on time for the presentations

4. Difficulty journal. Such document will allow us to keep track of diverse difficulties we’ve encountered throughout the project and the way we solved them in order to gain time if we ever encounter them again.

5 Conclusion

We, as a group, are enthusiastic about completing this project. Multiverse of Geekiness is a project that will require a lot of individual investment, in order to fulfill all our previously stated objectives, and to make it a genuinely entertaining game to play.
Finally, the project is only at its very beginning, which is why some slight modifications might be judged necessary or simply more pertinent. Such modifications (if needed) will be detailed in our presentations and stay within the initial tracks of this project.

6 Bibliography

Image 1 : from https://supermario-game.com/
Image 2: https://www.retrogames.cz/games/068/NES_01.gif