Project Worcestershire Fish, Second Defense report

Yvan Prevost, Eliot Champeil, Maxime Rampazzo

Raphaël Obled, Mohamed Lahbabi

Janvier 2025

Contents

1 Introduction

As part of our second semester computer project, we decided to create a video game : [Worcestershire Fish].

[Worcestershire Fish] is a game developed by Le Chalutier, our game development studio. It is a Metroidvania game where you control an egg that tries to evolve in a semi-linear world with a lot of surprises.

This document presents the progress of the project and the direction that it is taking.

2 Overview of the project

2.1 Reminder of the tasks

Tasks	Raphaël	Maxime	Eliot	Mohamed	Yvan
Website	substitute	responsible			
Music	substitute			responsible	
modeling		responsible		substitute	
Trailer			responsible		substitute
Game design			substitute		responsible
IA			substitute		responsible
online	responsible	substitute			
UI			responsible		substitute
decor modeling		responsible		substitute	

2.2 Tools used

The video game visuals are created using the popular drawing softwares Clip Studio Paint EX and Krita, which allows platform users to achieve professional-level

results, whether in animation or illustration. However, the software itself is not essential for creating the images crucial to the production of the animations present in the game. The graphic tablets used, a Huion Inspiroy 950P and an XP-PEN Star 03, adequately translate the vision of the artist in charge of the graphics onto the digital medium. This combination of software and physical tools provides an excellent way for graphic artists to create an environment perfectly suited to the conceptualized video game.

The music and sound effects (SFX) are all created using the paid software FL Studio, a music creation platform that allows the simulation of real instruments or the creation of synthetic sounds using plugins. This not only allows the composition of high-quality original soundtracks (OSTs), without ever worrying about running out of instruments, but also allows creativity to flourish thanks to the options allowing the user to create and use their own fully original sounds. In addition to FL Studio, Ableton Live is also used. Sharing the advantages of FL Studio, it is mainly used for live recording of real musical instruments.

The game is created and coded using Godot programming software, a free open-source application designed entirely for the video games creation. It is a general-purpose program, allowing the creation of 2D and 3D games, and offers a wide variety of platforms on which it is possible to publish one's game, from mobile to consoles. Programming can be accomplished using two distinct code languages, GDScript, which is exclusive to Godot, and C#. Godot being open source software, it is also possible to insert external modules and functionalities into the application, making

game programming easier.

For C# scripts, the software Visual Studio Code is also used to edit more comfortably the code. It does not make any difference in the end result whether Godot's built-in script editor or VSC is used.

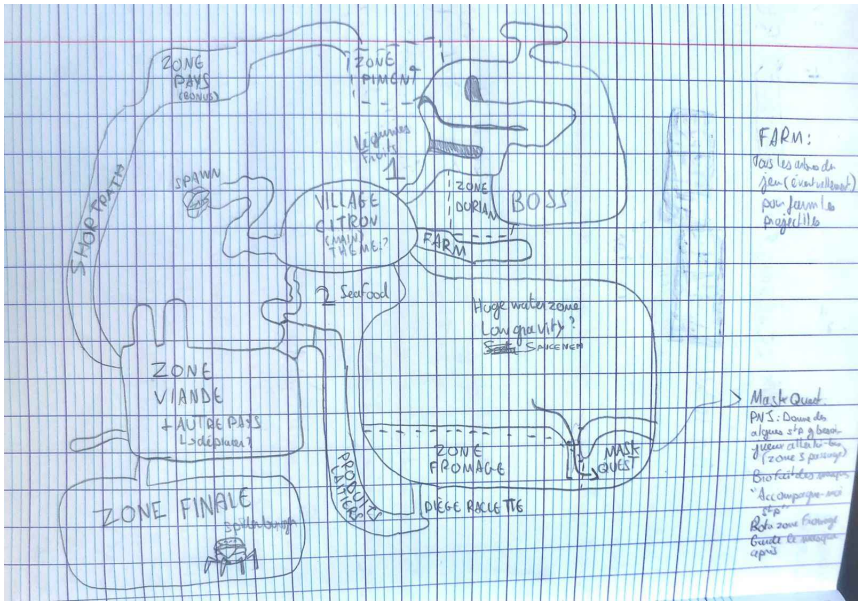
GitHub is a cloud-based platform for storing and sharing code written by one person to another. This feature is very useful, even essential, when a project such as creating a video game is undertaken by several people working in different locations and at different times. GitHub also makes it easy to make changes to code written by a team member during project design, which significantly speeds up the pace of work through faster, easier, and more optimized communication.

3 Project Progress

3.1 Game design

For the game design, we imagined many situations and scenarios, thinking about how we could utilize all the mechanics to create engaging gameplay.

Here is a potential map for our game:



For the game design, we also started creating some puzzles that we could include in our game.

3.2 Game mechanics

3.2.1 Movement

The player can now move left and right, as well as jump. The movement is handled via the manipulation of a vector of 2 dimensions titled Velocity. Each frame, the vector is updated according to the different inputs of the player. To do that, another vector titled Direction is used. A little bit of inertia has been added to make

the movement more fluent. The speed, jump speed, and inertia of the character is something that can be easily modified later in order to add features such as speed upgrades and ice physics. A tile map has been implemented so new walls and platform can easily be made and jumped on.

On top of that, the player now has access to a grappling hook. This allows it to move in a straight line in the air by latching on to terrain, which will allow us to have more diversity in level design and intended movement. This grappling hook will have 2 ways of being directed: one with the mouse of the user, and the other by rotating the grappling hook's direction using keys on the keyboard.

3.2.2 Camera

The camera node was implemented to the player so that the screen is always focused on them. This created a problem : if the player was close to the wall, ceiling or floor, the screen would show the inside of that wall, ceiling or floor. This is useless information for the player and takes space on the screen. To encounter this, we implemented limits to the camera so that it would not show further than a certain amount on the left, right and bottom. This allows for the player to always have the important information on the screen.

Issues regarding the camera behaving in unintended ways upon resizing the game's window have been fixed, and proper handling of the camera in multiplayer mode has been added.

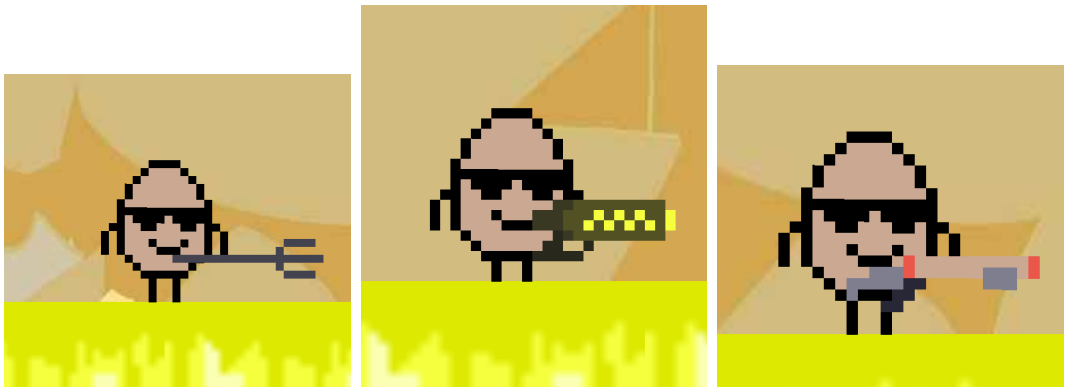


Figure 1: The three different weapons: Fork, Fry Gun and Hotdog Shotgun

3.2.3 Combat

Before, the project only included a fry projectile that was aimed with the user's mouse. Now, a diversity of weapons has been implemented to diversify the gameplay:
- A fork, acting in fashion of a spear, which will serve as a basis for potential additional melee weapons in the future
- A fry gun. This is a weapon that was made from the old projectile, with now having its dedicated weapon instead of coming out of nowhere from the player.
- A hotdog shotgun. This is a hitscan weapon that hits harder than the other projectile, but will come at a higher cost to the player.

With these weapons, the player will be able to fight the multiple enemies in the game. The player is only able to equip one of the available weapons at any time, though it will still have access to a default weapon if their weapon runs out of uses. It is currently possible to swap weapons in the shopkeeper's interface, which will be changed to a specialised equipment UI later on.

3.2.4 Environment

As stated before, the code for walls and platforms is almost complete. Additionally, the assets for one of the early regions, Cheeseland, were added to the game. The scrolling background repeats infinitely and moves at a different speed compared to the camera, as to create a sense of depth and perspective, called a parallax effect. The walls and floors use tile maps, which consists of using small sprites called tiled (usually square shaped) and repeat them in order to make environments faster. Physics were also added to the tiles so that the player can walk on them.

3.3 AI

The AI for a basic enemy has been finalised, this enemy starts chasing the player once they enters the former's chasing area, which consists of a circle around the enemy. When any object enters that area, the enemy checks if that object is the player. If that is the case, the enemy will enter chase mode. In that mode : he will 1) determine the player's position 2) determine if he is on the right or left of the player 3) move accordingly. When the player leaves that search area, the enemy returns to a passive state. The sprite for the first enemy of the game has been made and implemented.

Moreover a basic AI has been put in place for a panicking NPC (Non-Playable Character). This panicked NPC will start running left and right, switching direction at random intervals.

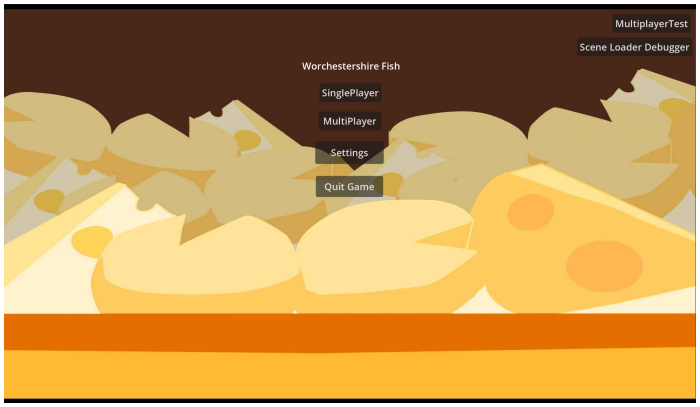
There is also a basic AI for NPCs in a general state of idleness. The NPC may either stand still, or move more or less randomly from left to right, giving the game

some life.

Both of these programs will serve to implement our future merchants, villagers and other characters that will animate the different areas of the map. We are planning to add a hub area were those NPCs will be particularly needed.

3.4 UI

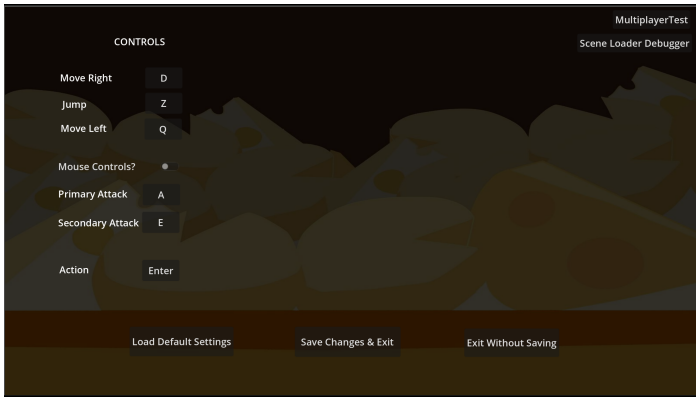
The starting menu is now fully integrated and functional. Furthermore, the players will now be able to interact with the basic multiplayer functionalities of the game, which have been made and are entirely accessible.



A basic main menu.

This starting menu now also includes an option to access the input settings, allowing the user to modify their game controls and allowing them to save their wished configuration in a file, which is currently located in a globalised path. Currently, the interface can only handle correctly characters from the alphabet, which will be

changed in the future to handle all keys of the keyboard.



Option selection screen.

Furthermore, the menu contains a simple "Quit Game" button, allowing the user to close the game. This functionality is only available in this menu for now, which shall be changed once a generalised pause menu is implemented.

The menu also currently contains 2 buttons used for debug purposes, which will be fully removed in the final project.

Regarding the In-game UI, the player now has a basic interface with basic graphic elements. This interface currently only displays the player's health and the money they own, which can be used in dedicated shops. Future elements will be added to this UI, such as the number of uses remaining on a given weapon, and a mini-map.

On top of that, the player now has access to a saving system, allowing them to maintain their progress in the game within a simple text file., located in the same directory as the control configuration file. Saving is only possible in specific areas of the map, at dedicated save nodes. The player is also capable of having up to two different save files.

3.5 Multiplayer

The multiplayer is now almost complete. To access it, the player only has to launch the game and arrive on the starting menu. Once there, the player can : generate their username, create a server, join one already created, and load the game. This allows the players to launch the test map by pressing a "Start" button on one of the connected user's interface. The multiplayer now has proper player synchronisation. However, it still requires some attention to properly manage specific events, such as dialogue boxes, user interface menus which must only be active on the correct player and not for all to see.

Restrictions to prevent any unintended inputs from any user's end, such as in the Multiplayer menu, beginning a game without having an existent server or starting the game as a client and not as the host, have been implemented. This is done by disabling the appropriate parts of the Multiplayer UI as long as certain requirements are not met.

The only major part of multiplayer that has yet to be implemented is a proper management of players disconnecting, be it through unstable connections to the server, crashes or voluntarily quitting the game session.

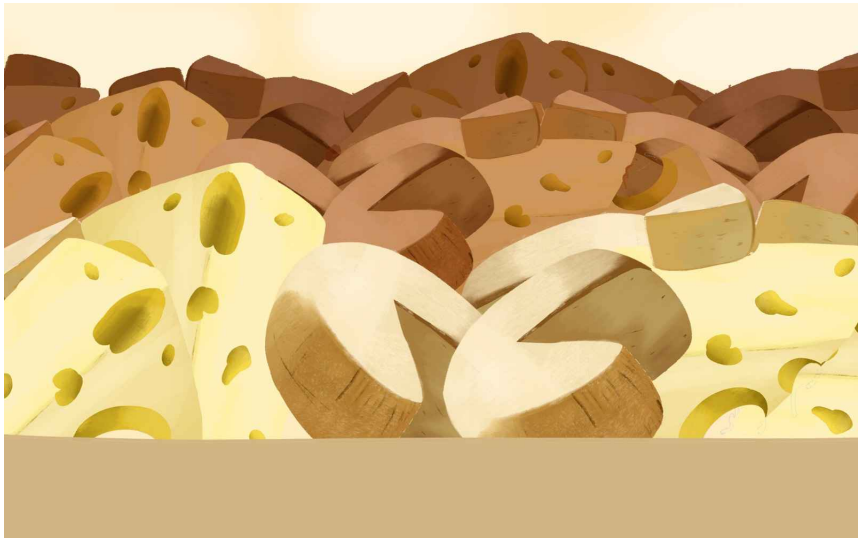
3.6 Texturing, Designing, Shaping

This part depends on the artwork made, therefore relevant information about textures and designs may be found below in the art section. However, there are

unique challenges faced with texturing and shaping, mainly making sure it all works properly. The textures drawn need to properly coexist with each other, and we have managed to incorporate an environment that looks and feels right with the tone we are setting for the game.

Since the last time, we have produced several designs and character animations for the game. Here are some examples:

The lemon shop



The second version of our cheesy background.

3.7 Website

For the website, we changed the design to make it more user-friendly and visually appealing. We also made the pages responsive so that the website works on any device.

Additionally, we added new functionalities, such as allowing users to download the game and some files.

Our current goal is to present our team and studio and to promote our game through the website.

To achieve this, we are also adding a page to showcase some of our design work and music, as well as a page for downloading the game.

3.8 Music

We also started creating melodies using physical instruments. In fact, we began composing and writing basic bass lines and guitar riffs that fit the game's atmosphere.

Furthermore, we have incorporated digital instruments into our work. Since the digital medium is still new to us, our current compositions are entirely made with digital orchestration, inspired by video game classics such as The Legend of Zelda and Hollow Knight.

Soon, we will begin integrating real-life instruments into our project. We already have some fully completed tracks for the game.

3.9 Art

The art of our project has been, for the most part, one of the more challenging aspects to execute properly. Animation is difficult, and even before that, coming up with ideas and designs requires strong imagination as well as the skills to bring them to life.

Since last time, we have created several assets, which has given us a better understanding of the artistic direction of our project.

4 Progress report

Tasks	Goal	Accomplished
Website	80%	70%
Music	60%	50%
Modelling	70%	30%
Trailer	50%	10%
Game design	85%	80%
AI	65%	50%
Online	90%	75%
UI	70%	60%
Decor modelling	80%	50%

The online progress is lower than anticipated, which comes from the delay before the previous defence and the challenged faced with elements such as user interfaces and proper synchronisation of complex elements.

Fields such as the AI have also suffered from less attention due to other fields being prioritised, such as the game's design and the UI. The ideas for behaviours of in game are plentiful, but their correct implementation can demand a lot of time.

The game's modelling and music have seen far less progress than anticipated, due in part to acquiring and learning to use the equipment necessary for their creation.

4.1 To do

Tasks	Final report
Website	100%
Music	100%
Modeling	100%
Trailer	100%
Game design	100%
IA	100%
Network	100%
UI	100%
Background texturing	100%

Here are the objectives that we have set for the last defense.

4.2 Features to be implemented in the future

For the website, we are going to explore hosting options and make some small adjustments. For example, we plan to add a section where users can watch the game trailer.

Regarding the overall game, we will soon be creating the world map, with a fully functional loading system. This will also include a dialogue system, and some lore to be discovered across the game.

For the design, we are going to produce more sprites and more general designs for our game and our site. We also need to produce some designs for our game's interface, which is currently bare bones.

For online functionalities, we will finish implementing all the features necessary for synchronisation across clients. This mainly concerns UI management.

For the music, we are going to produce more music to find the perfect music to use according to the game scenario and the gameplay. Our goal is to have a better immersion using the music. For the same reasons, we are going to create some

sound effects using several items. We're going to try to record real sounds for better immersion and good audio quality.

4.3 Possible improvement

We are going to make some slight modifications to the website's design.

This will make the website prettier, which is important to us because the web interface is crucial for future users who want to learn about our game or download it.

5 Conclusion

In conclusion, we have made significant progress by establishing the core functionalities, designing key visual elements, composing music, and developing the website. We will continue in this direction to bring the project to completion.