Project Worcestershire Fish, Defense plan

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 the project 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 software Clip Studio Paint EX, 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. Indeed, the graphics tablet used, a Huion Inspiroy 950P, adequately translates 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 the Godot programming software, a free and open-source application created entirely for video game 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 2 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.

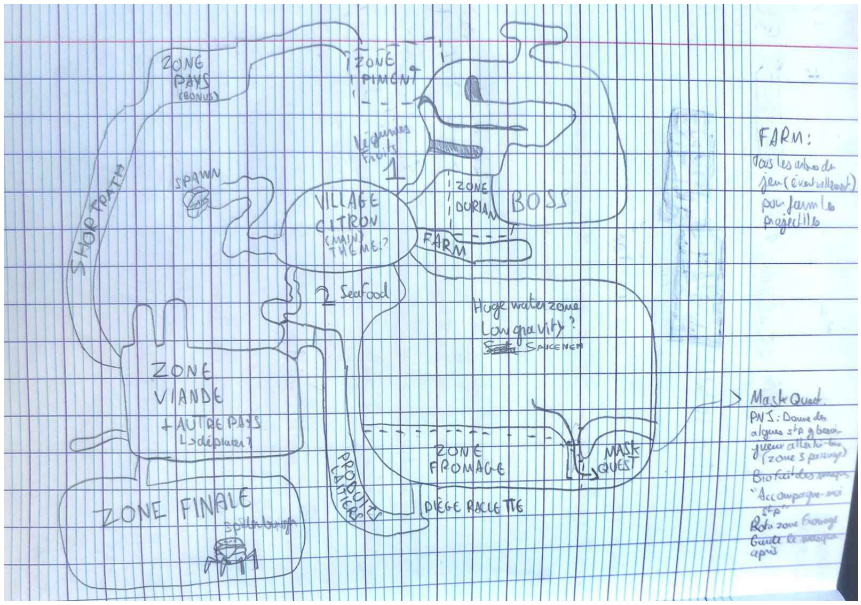
GitHub is a cloud-based platform for storing and sharing code written by one person with another. This feature is very useful, even essential, when a project such as creating a video game is undertaken by several people working in different locations and at different times. GitHub also makes it easy 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 started thinking about all the universes that the player can go through, and we thought about the specificities of each universe. We thought about enemy specificities and unique items for each part of the game.

Here is a potential map for our game:



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

3.2 Game mechanics

3.2.1 Basic 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.

3.2.2 Camera

The camera node was implemented to the player so that the screen is always focused on him. 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 farther than a certain amount on the left, right and bottom. This allows for the player to always have the important information on the screen.

This camera is not perfect however, as going int full screen with the game window extends its range of view compared to if the prototype was running at its default window size. These issues are currently being worked on.

3.2.3 Shooting

The code for a projectile has been implemented. The player can now shoot basic objects that delete themselves after coming into contact with other objects (enemies, floor, walls, etc.), this will allow us to make different projectiles with different sizes,

speed and damage. In addition, the elements that collide with the projectile can themselves detect the hit and respond accordingly, which will allow an easy implementation of a health system later on, and even destructible environments. The projectile determines the vector of his trajectory by subtracting the position of the player from the position of the mouse.

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.

The scrolling background also suffers from putting the game into a full screen window, which causes it to no longer cover the entire background.This issue is currently being worked on as well.

3.3 AI

The AI for a basic enemy has been finalized, this enemy starts chasing the player once the later enters the former's chasing area. The chasing area consists of a circle

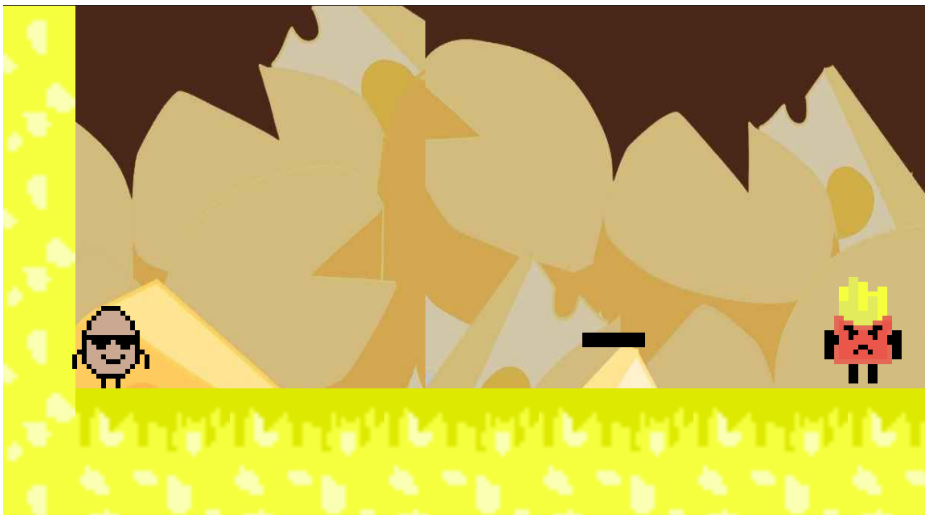


Figure 1: first look at the game :
the main character, an enemy and a bullet (in black)

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.

3.5 Multiplayer

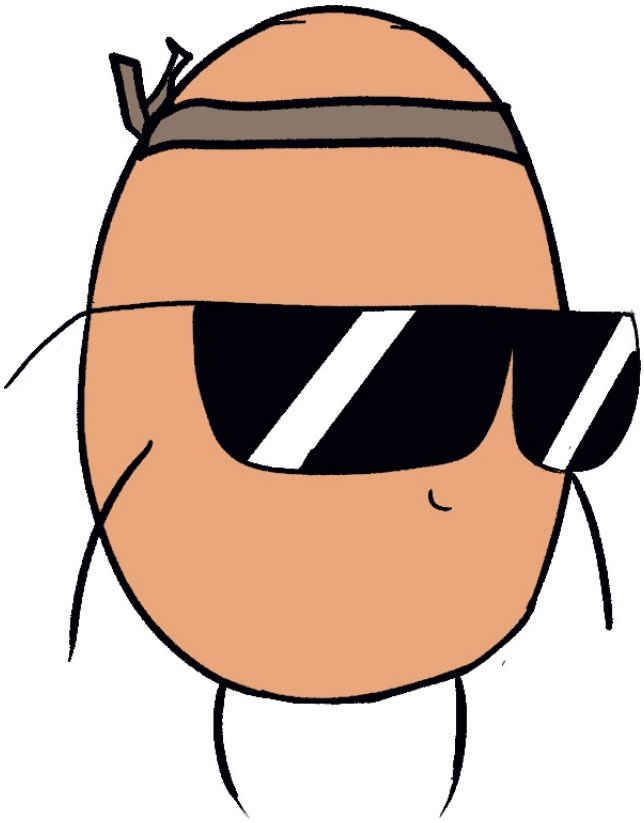
The multiplayer is now semi-functional. To access it, the player only has to launch the game and arrive on the starting menu. Once there, the player can : generate their

username, create a server, join one already created, and load the game. Currently, this only allows the players to launch the test map by pressing a "Start" button on one of the connected user's interface. The server is already capable of keeping track of which user should be associated to a certain character. More functionalities, such as player spawnpoints, synchronisation and input restriction to the correct player shall be added soon. 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, will also be implemented in the near future. This will be done by disabling any parts of the Multiplayer UI as long as certain requirements are not met.

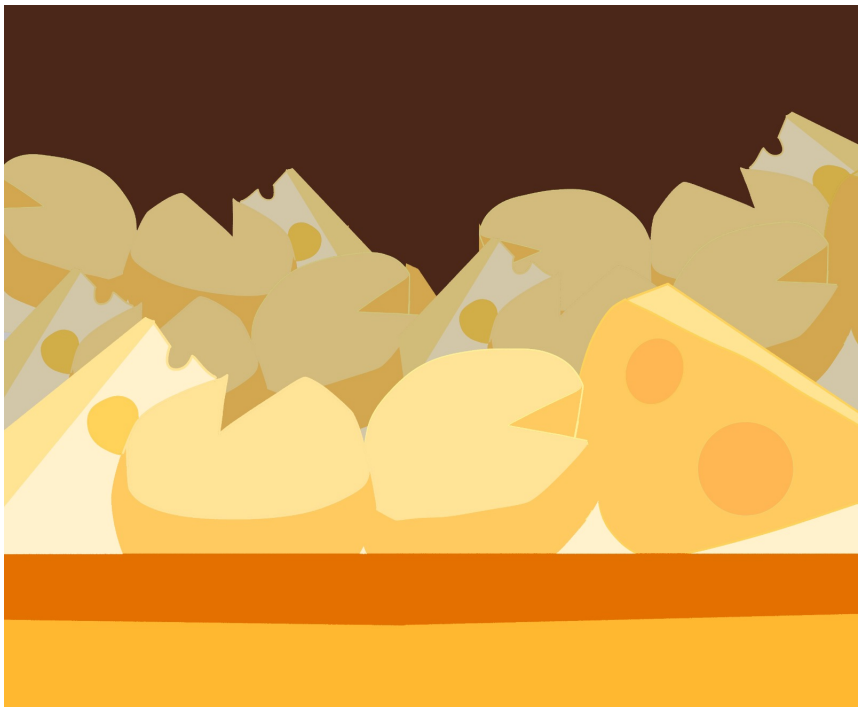
3.6 Texturing, Designing, Shaping

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.

The current prototype contains the background below, as well as a running animation for the player's character. Characters have the possibility to alter their animations by flipping them based on their movement direction, and if they are performing certain actions. Multiple placeholders have been used for the existing NPCs and map, which will be replaced in the future with their definitive versions.



”Coullègue”, our egg



A cheesy background

3.7 Website

For our website, we have built a basic version showcasing what we are able to do right now. It includes a general overview of the project, an introduction to the team as a whole, profiles of each team member, and some sprites that are already ready.

Our current goal is to introduce our team and give a preview of what the project will become.

To do this, we are also adding a page to share some of our design work and music.

3.8 Music

We also started to think about some melodies with physical instruments. In fact, we started to find and write some basic bass lines and guitar riffs which correspond to the game. Furthermore, we have also incorporated digital instruments into our work. Seeing as how the digital medium is unexplored, our current music is entirely composed on digital orchestrations, inspired by the classics of the video game medium such as the Legend of Zelda and Hollow Knight. We will soon begin to incorporate real life instruments into our project.

3.9 Art

The art of our project has been for the most part one of the tougher aspects to properly execute. Animation is hard, and even before that, ideas and designs demand powerful imagination to come up with, as well as the skills needed to bring them to life. We have managed to work on this aspect, as we do possess placeholder assets and an animation. However, while it still needs polishing in order to be considered complete, it proves that the capabilities needed to ensure a high enough quality for our project.

4 Progress report

Tasks	Goal	accomplished
Website	45%	35%
Music	30%	30%
Modeling	30%	20%
Trailer	10%	10%
Game design	50%	50%
IA	15%	15%
online	50%	35%
UI	30%	30%
decor modeling	40%	40%

For the website, we did a little less than we were expecting. When we set our goals, we underestimated some things that we could not implement yet. For example, we do not have all the designs, so we cannot put a lot of design elements on the website. We also do not have all the documents that we can upload to the website.

The online functionalities of the game have seen less progress than anticipated. This is due to the new aspect of synchronizing data between the different clients, whose difficulty we had underestimated.

The reliance on Godot's built-in C# script editor also caused delays, as it is not very user-friendly and forces reliance on the official documentation. This can be harder to understand compared to attempting to use appropriate methods with outside coding software, such as Visual Studio Code which gives easier to grasp feedback in case of method misuse or attempting to learn to use a new method.

Implementing restrictions to ensure only the intended client has impact over certain elements, such as character control, proves to be challenging as well.

4.1 To do

Tasks	Second report
Website	80%
Music	60%
Modeling	70%
Trailer	50%
Game design	85%
IA	65%
Network	90%
UI	70%
Background texturing	80%

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

4.2 Features to be implemented in the future

For the website, we are going to add a space where we can download some documents, such as the CDC's or this document. We are also going to enhance the website

as well as making it more personal, using some designs which references our game and our studio.

For game mechanics, we are going to add more enemy types, more weapon types which implies more projectiles, and finally more areas. In the immediate future we are going to work on finalizing Cheeseland and making a grappling hook.

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 synchronization across clients. This includes limitations for clients to act on their characters only and interactions between players.

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're 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.

For the game design, we are going to find new puzzles and new enemies that we can add to the game for the best user experience and to have a more diverse game.

4.3 Possible improvement

We are going to redesign the website using the ones we've created.

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.

For the game design, we are going to create a final map by selecting from all of our ideas. The goal is to have a clear map and to know what we should implement and what we should not.

5 Conclusion

As a conclusion, we have put in place the basis of the functionalities as well as realized the basis of the designs, the music or even the website, we are therefore going to continue in the same direction to carry out this project.