Bartendu

First report

Jean Bou Raad
Lou Lefebvre
Vincent Thirouin
Emeline Tichit

EPITA 2024

https://areas0.github.io/website/

Contents

Introduction

This first report will serve as a follow-up for the book of specifications, as well as a description of the tasks that have been carried out since the beginning of the project. Difficulties encountered and chosen solutions will be discussed.

As a reminder, *Bartendu* is a cooking bar simulation game in the making.
Les Cosmopolitains have chosen to undertake this project in C# on Unity. Other software and tools have also been carefully selected such as Blender.

The goal for players is to make drinks and cook to satisfy customers in a bar. Room (restaurant) management is also at the core of the gameplay.

Therefore, a lot of tasks must be done on two main sides: technical and artistic.

For this first stage of the project, we aimed to produce the basics for the game, meaning getting a working multiplayer mode, creating the models for the main pieces of furniture and the items, as well as the soundtrack. We also finished the core mechanics of the game and the website has been updated since we handed the book of specifications in. Basically, we worked on the most prominent features.

1 General progression

Task	Expected	Completed
UI graphics	5%	20%
UI implementation	10%	25%
Character actions	90%	90%
Multiplayer	90%	90%
AI	10%	8%
Tutorial	0%	0%
Level design	15%	15%
Level implementation	10%	15%
Object implementation	5%	5%
Particle Effects	0%	10%
3D modeling	15%	15%
3D animation	0%	5%
Music	10%	20%
Sound Effects	0%	0%
Writing (Story-line)	0%	0%
Website	80%	80%

2 Additional details

As for everything in life, humans tend to bite more then they can chew. We are only human and did not escape that idiom. But we gave all we had and manage to stay approximately on track. Most of what we had plan got properly done. There is however some aspects of the project where we had to give up some ideas in order to deliver a full and polished project. We did not specify how many levels we planned on implementing in the game but their addition to the game impacted the advancement board.

This is why we are well ahead in terms of UI and UI implementation because of the amount of levels going down. We also did not expect the game to take that long as we did not take in consideration all the different bugs and obstacles we had to overcome. That is why we did not have time to work as much as we wanted to on AI. Despite having a good amount of research done on the subject, we could not start implementing it as the game was not yet finished at that time.

With all this explained we still would like to stress that not a lot of concessions were made and that we are, for now, on track with the project.

Furthermore, we are ahead of schedule in terms of particle effects and 3D animations as we realized that we needed to start working on those parts earlier, to make more room for other aspects of the project in the upcoming period.

3 Task distribution

We all are expert in a particular domain so we thought it would be best to divide the work by our areas of expertise.

Jean being the leader of the group has a big role that is coordinating everyone's work. Being in charge of the level design and the story mode means having to join all together programmation, 3D assets and details like particle effects, sound effects and animations. Not using his huge talent in the art of music would be an incredible loss... That is why he also is responsible for making the music of the game. Finally, as he had some experience with web development, we decided to leave that task to him knowing that the job would be perfectly well done.

Vincent has been programing for the longest and thus has a lot of experience. It was a wise decision to let him handle this as he knew his way around by having already completed many games in the past. A huge part of his work will be to work on the AI as one of the main goal would be to use reinforcement learning to get different levels of artificial intelligence. He also became responsible for creating the sprites for the UI as well as the logos. Unity being a software that he has been using for the last 3 years, he also is in charge of implementing everything in the game from musics to 3D models.

Lou is in charge of 3D modeling, which is the creation of all the assets that will be used in the game. She is a very creative person and someone that takes her work very seriously; that is why she is a good fit for handling the design of the different assets. Moreover she is very good at working with people which makes the collaboration and the creation of assets that will match the levels that Jean imagined easier.

Emeline is mainly in charge of writing the reports. Her hard work and sense of perfection always makes her work stand out. She proved to be very efficient for the book of specifications by reformulating everyone's part to make a coherent report. Apart from writing latex, she also has to create the final touches that will make the game perfect. That includes particles effect such as fire, bubbles in the sink or even the extinguisher's smoke. But also sound effects to make the game more alive (like the drizzling sound of bacon, the sad sound of defeat or the cute sound of mice trying to steal food). Lastly animations will add a final touch : the cherry on top ! She will later have to write a tutorial for new players.

4 Achievements

4.1 Multiplayer

The first thing we started working on was the multiplayer. In all honesty, implementing the very first version of it was not too hard; we already had experience with it, and in the course of a weekend, we already had a "working" multiplayer. Functional, but not final. Players could join a lobby where all the rooms were and then could click on one room to join it and then load the correct scene. Nothing fancy but very efficient. We would later come back on the multiplayer to add matchmaking. Matchmaking will be used to matchmake a player with the correct room in the lobby: for example, a player who would like to join a player against players would not be able to see a room where the game mode is players against AI. This was the last time we worked on the multiplayer. There is still some tweaking to do for user-friendliness and matchmaking, but those changes will only take place during the period between the second to the third defense.

4.2 Game mechanics

4.2.1 Player movement

Once the multiplayer was ready, we quickly started working on the game's mechanics, knowing how much work had to be done by the first defense. Player movement was first and was a breeze to implement. We used physics-based movement to make sure collisions would happen properly. Four actions were implemented: the first was one being able to take or drop items, the second was being able to use an item (for example a cutting board or the extinguisher), the third action was being able to throw an item and lastly we implemented the action of boosting to allow the player to have a small boost in the forward direction.

4.2.2 Spawning items on the server

This was the very first problem we encountered in this section: the ability to spawn items. We were first working on a single computer for this, thus it was very hard to test the multiplayer and was very time intensive as we had to build a version of the project every time. When we started to develop the game on two computers, we had the very sad revelation that spawning an object on a client's side and then synchronizing its position was not working... We then had to find a way to instantiate an object on the server's side which took us a while to figure out. A normal instantiate returns the object instantiated but a Photon Network instantiate does not, thus making it hard to get the object and set its parent... We had to create an object handler class to react as soon as the object was instantiated and to catch the data sent while instantiating, so that it could find the photonID passed and set its corresponding gameObject to be its parent.

4.2.3 Adding core functionalities

We then added very essential functionalities in the course of about 2 weeks. We will only quote them as we had no difficulties implementing them.

We created a script SnapToSurface that is pretty self-explanatory, it just snaps to the center of the surface so that we could maintain a grid-like system. Then we added the useSurface to be able to use a surface such as a cutting board or a sink. With those 2 scripts we were able to create a cutting board, a work surface, a sink, a trash can, a table, chairs, and much more. But then we started working on moveable surfaces and this was probably the hardest mechanic we had to work on.

Moveable surfaces are essential to the game because they represent plates, pans, glasses and more... Again, it is straight forward, it is a surface (that can hold an object) but also an object (that can be moved). The huge problem was to move the 2 objects at the same time, and this caused all sorts of weird bugs. The player would start to fly, a giant tomato would turn into a huge disk, players would clip through the floor, objects would not follow the intended holder... All these bugs took us almost a month to figure out. And working on plates for a month without knowing what caused the problem is very frustrating... The problem turned out to be collisions and physics messing up the object's scale and doing all sorts of weird stuff. After that huge mess was cleaned up came the time to start working on the room.

The goal here was to implement a client generator capable of generating clients with orders for the player to complete. This was a whole new problem to solve because this time it was not a technical problem but more like a level designing problem.

Came the time to wrap up the whole thing. We created a level controller capable of handling the start, end, score and timer of the level making the base game and core mechanics finished. All we have to do now is to add as much details, levels and items to make the game perfect.

4.2.4 Implementing assets

While implementing the different assets in Unity we had to make sure they fitted the size of the level and were consistent throughout the game. We had a lot of problems and had to work together to fix them. We had to create the different levels, and then with the first versions of particle systems (fire, extinguisher and water) and the music, there was a lot to do and experiment with. The good thing is that we now know how to handle those assets and their implementation will be faster next time. Lastly, we implemented some sprites that were created for the game and created a small menu.

4.3 Creating sprites

We really had no experience creating sprites, but doing it was quite entertaining. Before submitting the book of specifications, we created a first logo for the company and a logo for the game. Then we started to create the sprites for the game that will be used for the orders and the display to know what the item is made of (for instance, a burger with bread and a steak will display the bread and the steak sprite).

4.4 3D modeling

3D modeling is very important in this project because the game has a lot of objects and assets. Since we want a game that looks unique and like we imagine it we decided to create each asset instead of using some that can be found online or in open sources. This is why, to create those assets, we have decided to use Blender.

Some assets that we have already created are :
— Ragdolls
— Tomato / slice of tomato
— Lettuce (Fig. 1) / lettuce leaf
— Extinguisher (Fig. 2)
— Steaks (burned, cooked and raw)
— Classic glass
— Pan
— Bread
— Plates
— A Hamburger and all the different possibilities to assemble it (Fig. 3)
— Salad (Fig. 4)
— ...

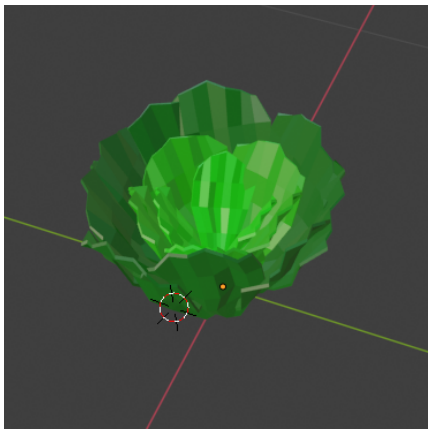

Figure 1: Lettuce

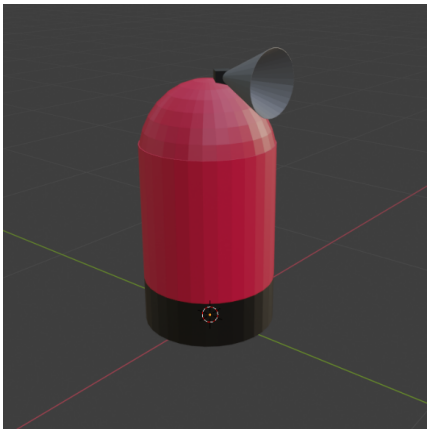

Figure 2: Extinguisher

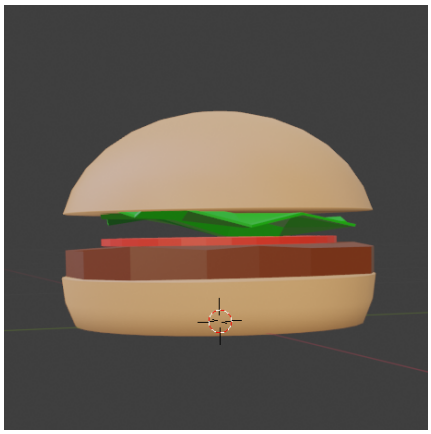

Figure 3: Hamburger

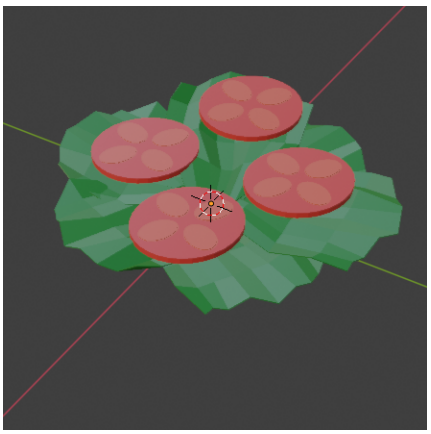

Figure 4: Salad

Those assets were the first that were created because they will be needed in most of the levels. Moreover to create those objects, we needed to add colors and we plan on adding texture later on (for the glasses,...).

Then some assets (like the ragdolls) will need to be animated: in order to do that we had to add bones and weights to the assets so that the animation can be done. (Fig. 5)

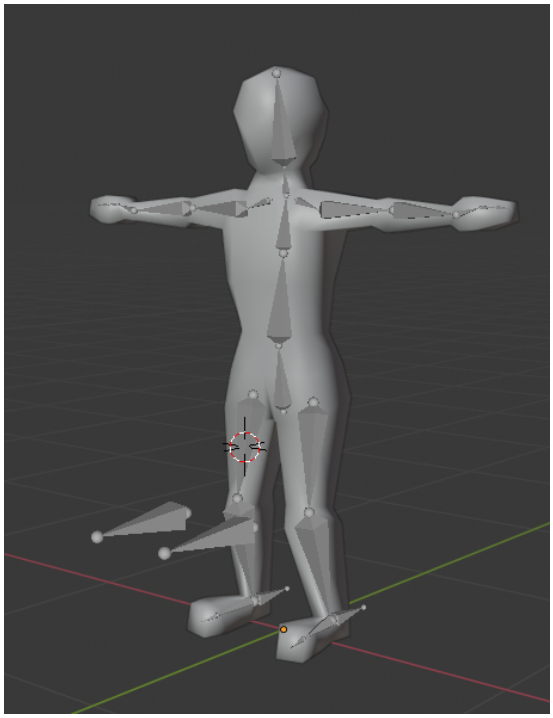

Figure 5: Ragdoll

Using Blender was not as difficult as we expected but we encountered a few difficulties, especially to export and import the assets on Unity:
- The rotation of the asset and its location were different, so we fixed them by applying the rotation on the object
- The scale and size were also different in these two software (we used the same fix as the rotation)
- Finding the correct mesh and form to obtain what we needed
- Some faces were invisible on Unity but not on Blender, so we had to solidify the faces that were "invisible". This consisted in adding a transformer to solidify and add a little volume to those plans.
- The colors in Blender and Unity seemed different so we added a light in Blender to choose the color and then removed it
- ...
Then to export the assets we transformed it as an FBX file and checked the box "Apply transforms".

There were other difficulties with using Blender because we had never used this software before:
- Figuring out how to add colors on different faces
- Learning the different shortcuts and how to use the different tools in Blender
- Figuring out how to assemble different elements
- Learning how to add bones and weight to an asset
- ...
We eventually managed to understand how to properly use Blender by spending time on it and thanks to some online tutorials.

The group is currently on schedule concerning 3D modeling. Thanks to Blender we will be able to create any asset we may need for this project and new ones will be done for the next presentation.

4.5 Particle effects

We started working on particle effects using Particle Systems. Those are made directly from Unity and are especially useful to represent chemical reactions, fluids, smoke, etc... Anything that is not a solid and hence cannot be modeled in Blender in the same way as items and furniture. The first versions for fire, fire extinguisher smoke and water were implemented. However, we are still investigating whether we should use shaders instead, as we found that for some of the effects, like fire, these tend to look much better. The main issue with this is the compatibility with the Unity template that we are using for the game, which is the 3D template. We would need to install some packages which would probably cause some conflict with the graphics we have already implemented.

Figure 6: Water

Figure 7: Fire

Figure 8: Smoke

4.6 Music

For the first defense, we have aimed to create the necessary music for the things we have already developed. Therefore, we have written three original partitions: one for the main menu of our game, another for the first level, and the last one for the rush period of a game on this level.

Creativity difficulties aside, the creation is quite easy. Indeed, Musescore, with the help of a midi keyboard, makes writing very easy because the layout is done automatically.

Here, you can find an example of the quality of layering of this powerful tool:

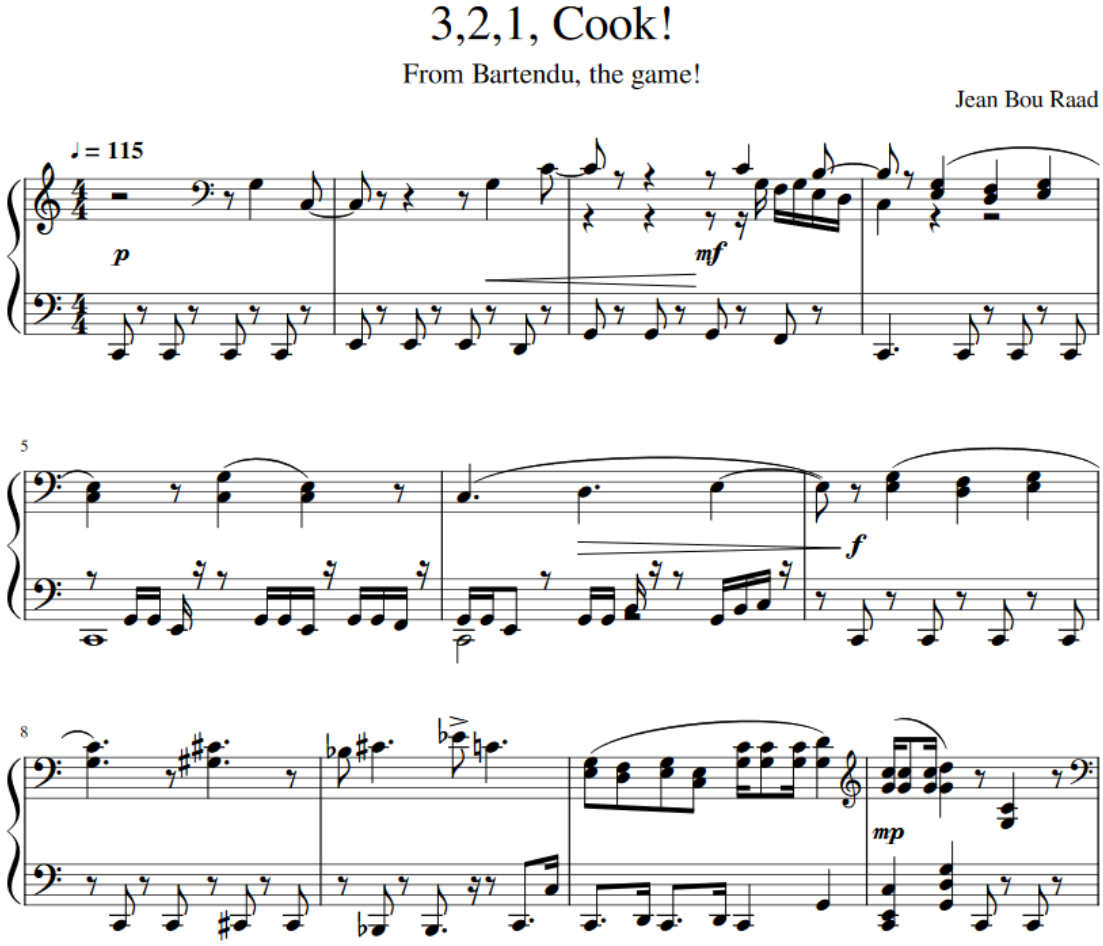

Figure 9: Music sheet for one of the songs included in the game

However, the music sheets are not enough to create a sound of good quality: their output is raw. And to do better, Ableton Live 10 can take as input the sheet as a MIDI file and then gives plenty of options and instruments to create good quality output. But it requires more work. Moreover, some instruments are not available in the basic package, and it limits creativity.

We will now explain the details regarding music creation. For the main menu, we have chosen a plain kind of nostalgic style inspired by the *Life is Strange* main menu soundtrack[1]. Then an acoustic guitar was mandatory. The piano was also chosen. Thanks to the experience of some members as pianists, we can easily write sheets for this instrument. Going with the nostalgic approach, it was clear that tempo should be slow and a minor scale. But an only nostalgic approach would have been too sad for the game that *Bartendu* aims to be, so we brought some light to the theme at the end by going on a major scale and with a more rhythmic approach. As for all games, the track is a loop that never ends until the player does something else.

For the in-game music, it was obvious that we had to write something dynamic and swinging. To do that, we have chosen only piano instrumentation which offers great flexibility. The main loop has a fast tempo (we would call it in music *Allegro*). With that in mind, offbeat was also used to create some tension for players. For the melodic approach, we did not have any major source of inspiration except for a small part inspired by the *Dialga & Palkia battle theme* from *Pokémon Diamond Pearl Platinum*[2].

For the rush loop of this theme, we wanted to reuse some parts from the previous theme while also creating more tension. Hence, the tempo is going a little faster in some parts. However, the result is not satisfying and it might be reworked later.

4.7 Website

A website has been created by the team to show the progress of the project. And it also allows downloads for various files such as executables of the game itself and some documents.

The website is available at the following address: https://areas0.github.io/website/

4.7.1 General features

The website is a key element to help people get to grips with our project and discover the game.
Therefore, we have built five pages gathering essential information about the project. That includes a home page with a news zone updated by the team, a download page with the executable of our project, and some documents such as the book of specifications and the first report. There are also pages about the team and credits for the software used and schedule for the project.

4.7.2 Backend

As mentioned before in the book of specifications, the backend is handled by Github Pages which provides hosting for the website, server backend, and error management. The only requirement to put the website online was to make it accessible in a Github repository available at https://github.com/areas0/website/.

[1]Life is Strange main menu soundtrack: https://www.youtube.com/watch?v=d9ENy1v3Dyg
[2]Dialga & Palkia battle theme: https://www.youtube.com/watch?v=I_57ptO3TKc

4.7.3 Frontend

The website has been made using classic HTML language and some CSS for theming and layering.

The frontend was quite long to develop. Indeed, we chose to build a website without the use of any template and only with the use of a well-known web toolkit: Bootstrap. Bootstrap allowed us to make a good layout for the website with the help of a system of rows. These rows are also key for the cross-platform compatibility (PC to a smartphone) but also were quite hard to master. Thanks to the help of many examples on Bootstrap's documentation, we were able to make a responsive website that adapts well to all types of screens.

The theme developed is classical with a dark-grey background and some colors which fit the graphical charter of our group.

Below, you can find a screenshot of the main page of the website:

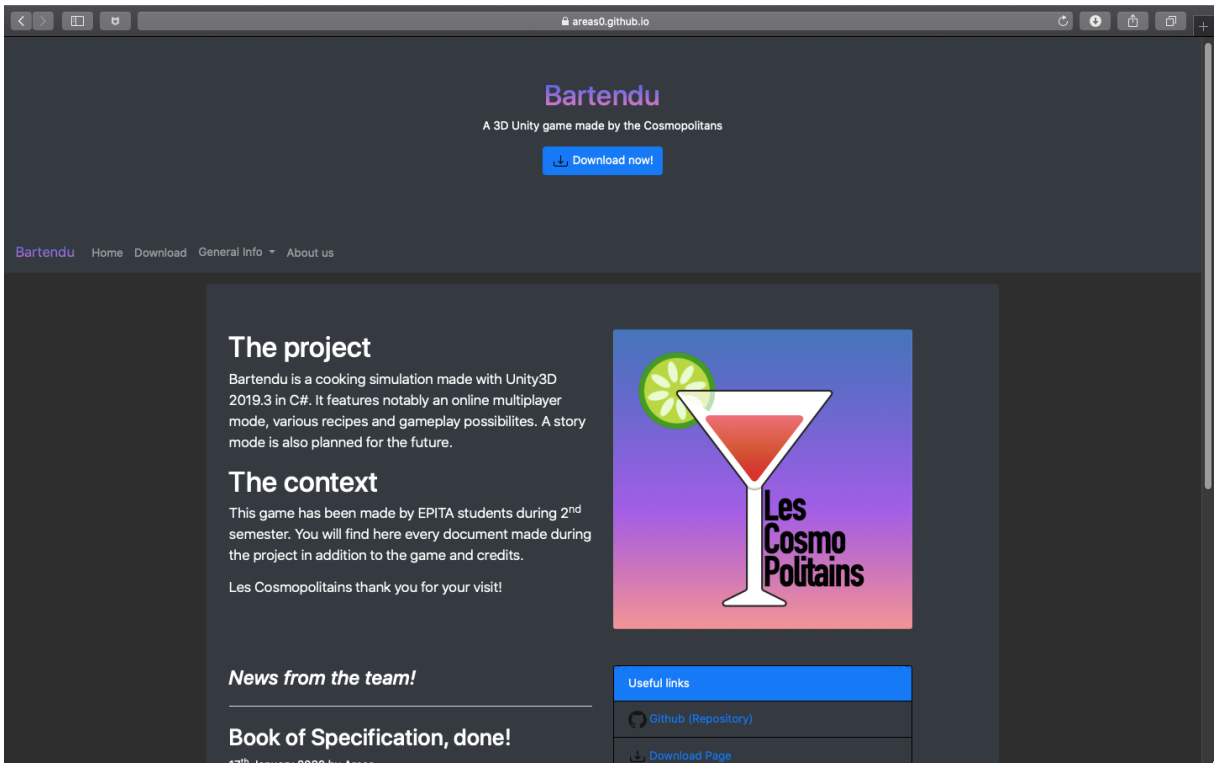

Figure 10: The home page

4.8 Level design

Bartendu is nothing without its levels. Therefore, this task has been started early with two subtasks: level creation and implementation.

4.8.1 Level creation

For the first defense, only one level has been properly implemented in-game because all mechanics were not available yet. However, three levels have been created with their specific gameplay possibilities and environment.

The first one is a simple kitchen to restaurant zone level, which is the base of what we aim to create as standard gameplay for *Bartendu*, and its implementation (detailed later) has been done in the white box stage.

The second level is a level on a boat. It uses surface available to create a fully capable restaurant and dynamic gameplay. But it will require more work than other levels to implement fully because it requires many assets and new recipes.

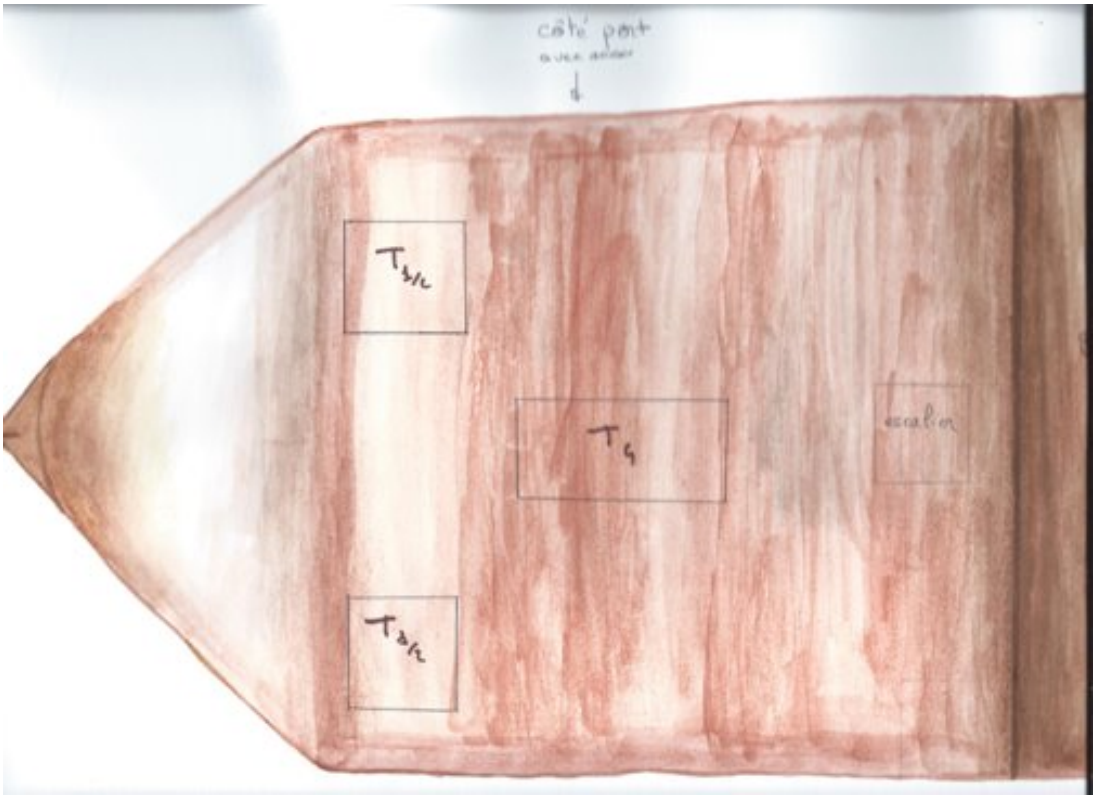

Figure 11: Sketch 1 for the second level

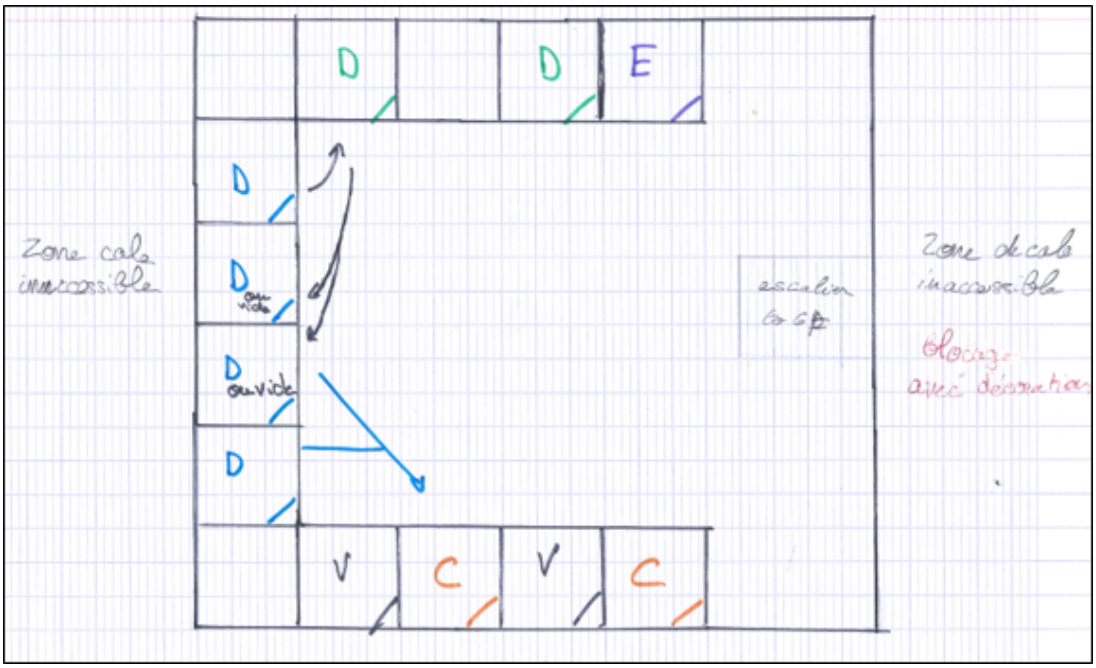

Figure 12: Sketch 2 for the second level

Key to read the sketches:
D block: dispenser for food
V block: empty zone to exchange plates
D block: cutting board
C block: hotplate
T block: table with a given number of customers allowed
E block: sink

Finally, we have created a level in a school environment. The cooks will have to work in a school environment with a teacher writing at the board. And if he stops writing, then the cookers must stop cooking too. It will require help from various team members because many assets are needed, and some scripting must be done.

4.8.2 Level implementation

Following the schedule, only one level has been implemented in the white box stage. It will allow testing of gameplay and will allow enhancing the game.

The team has chosen to work with a tool on Unity for level creation named ProBuilder. It allows us to create a simple level with simple geometrical forms and integrate assets later (e.g. polished 3D assets) more easily.

Using the first assets built by the team, we have created a gameplay loop illustrated below:

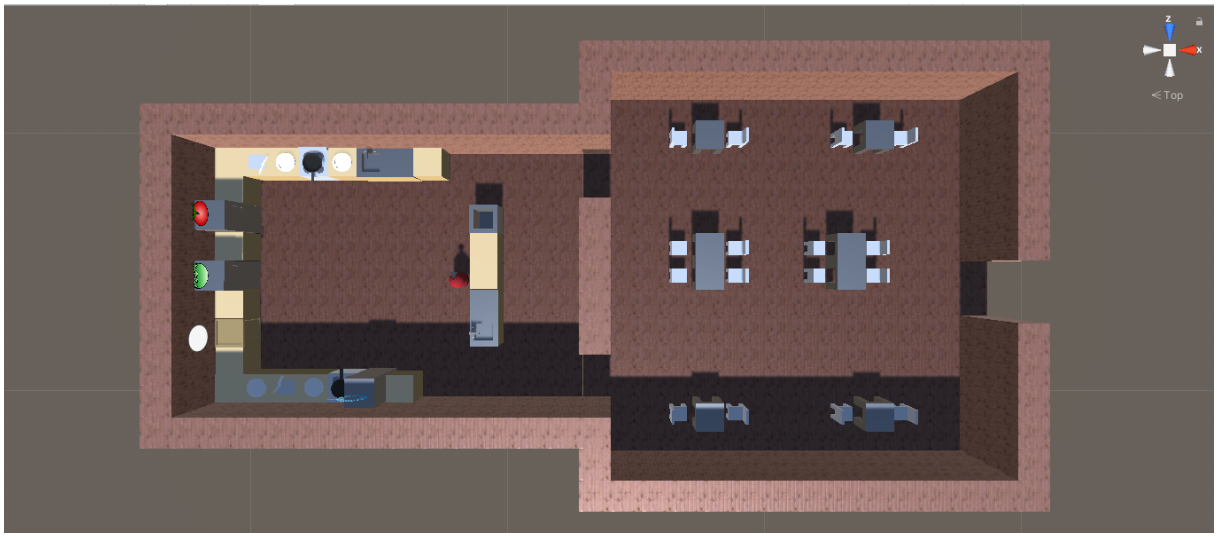

Figure 13: A screenshot of the basic kitchen level in Unity scene view

This level was built using a symmetrical axis in the kitchen to ease players' organization. After taking ingredients (zone at the far left), players can cook these using cutting tables and hotplates. After that, they are put all together on a plate and centralized in a zone next to the customers' area. Then the player that has cooked the plate can choose to serve its customers or can wait for its team to do the job. Players will also have to handle the clients by taking them to their table. Currently, it does not have the bar part. The main reason is that it was not ready yet when the level was implemented.

4.8.3 Order generation

To make entertaining levels, the levels' designs are essential but not enough to create an entertaining experience. We created a system of order generation.

During a game, players will have to deal with different kinds of customers because they have different requests. Thus, we have created some rules for how we will generate those clients.
These rules are:

- No more than 4 orders at the same time: players must have some pressure but not too much because it will just have a negative mental impact.
- Each set of order will be assigned to a group of 2 customers. They are considered as one order

There will be two successive draws:

1. The number of customers' requests to generate (2 or 4 customers: 1 or 2 orders) chosen randomly under certain conditions (see the above)
2. The dishes requested are then chosen randomly with equiprobability between each of the recipes. The same goes for the drinks.

Once it is generated, the request has a timer defined by the recipe asked. The timer has a static value for each recipe (no random). Finally, the generation procedure is called following a hidden timer that will fill fast the queue if it is almost empty and if it is full will give time for players to fulfill the already existing requests.

The group is currently on schedule for level implementation. *Bartendu* offers a large universe of concepts and gameplay possibilities to exploit. With the current tools in Unity, the implementation in the white box stage is fast. It will allow us to perform well for the next deadlines.

Conclusion

Despite some difficulties which have successfully been resolved, the team is currently on schedule concerning all tasks. This means that about one third of the project has been done.

To summarize: the website is online and active, the first musics were composed, the most important assets were created and we all got used to using the different software. We created a solid base for the game with all essential mechanics and features, including a working multiplayer, where we can expand from until the last defense.

We aim on pursuing in this direction for the next defenses; we still have a lot to do for this project such has implementing a working AI as well as the single-player mode and we also would like to add as much levels as possible along with refining details and playability.

Finally, we are aware that we still have a lot of work to do but so far this project has been a really good experience for all members and has brought us closer as team as well as allowed us to learn new skills such as the use of different software.