

Bartendu

Book of specifications

Jean Bou Raad
Lou Lefebvre
Vincent Thirouin
Emeline Tichit

EPITA 2024



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

Contents

- Introduction** **3**

- 1 Les Cosmopolitains** **4**

- 2 Presentation of the game** **6**
 - 2.1 Bartendu 6
 - 2.1.1 Concept 6
 - 2.1.2 Levels 7
 - 2.1.3 Game design 7
 - 2.2 (Cooking) Simulation games 8
 - 2.2.1 History and definition of the genre 8
 - 2.2.2 Point of this type of game 9

- 3 Production** **10**
 - 3.1 Tasks distribution 10
 - 3.2 Description of the tasks 10
 - 3.3 Schedule 12
 - 3.4 Tools and resources 13

- Conclusion** **13**

- Sources** **14**

Introduction

This book of specifications will provide an overview of our project named *Bartendu* for the second semester. This game will be made by Les Cosmopolitains in C# on Unity mainly with the help of other tools such as Blender. (to be detailed later)

Bartendu will be a cooking bar simulation game. Players will cook and make drinks to satisfy customers in a restaurant. This idea came mainly from the experience in this kind of game Vincent and Jean had. But this was not enough to make a good idea. We thought that cooking was not enough, we wanted something more: drinks recipes and room (restaurant) management to make it even funnier.

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

Indeed, the group aims to provide a multiplayer experience around many levels designed by us. That means implementing various gameplay possibilities with recipes, environment manipulation. Once that first experience is polished, we aim to create a co-op story mode that includes restaurant setup editing, in-engine cinematics, artificial intelligence. All these tasks are detailed in this book of specifications along with the main artistic tasks to make a cartoonish-style environment supported by a musical environment. Each task is assigned to a group member with a substitute.

Everyone will experience coding to make a game on Unity. All team members will get a chance to experience game designing, creating a musical environment and 3D assets making.

Finally, you will find in this book of specifications a detailed description, approach, and schedule of all these tasks with the deadline for each of the presentations.

1 Les Cosmopolitains

Led by Jean Bou Raad, Les Cosmopolitains is formed of 4 students of Eng2. The name of the group, which originates from the diversity of its members, is also a nod to the famous Cosmopolitan cocktail. This is a direct reference to the game we plan on creating, *Bartendu*.



The team:

Jean Bou Raad (Project Leader):

Computer science has always been a passion for me even before EPITA. I have learned by myself a little bit of some programming languages and got the opportunity to work with a small team for server software development in C# and node-JavaScript (known as node-js). I got in touch with various developers and discovered the world of software development with them.

However, I have no experience with Unity. Therefore, I aim with this project, to learn how to make a game with the help of the rest of the team. Furthermore, I see this project as an opportunity to expand my skills in sound environment and music-making.

As the project leader, I will put my organization skills at test during this semester to make this project a reality and fulfill our objectives.

Finally, as a future engineer, this project is an opportunity to learn how to make a team work.

Lou Lefebvre:

I always loved computer sciences but I never actually learned to code before coming to EPITA, even if in one semester I learned a lot.

However, I already did some projects in SI (high school) and with a innovation group with whom I had the opportunity to present projects to different competitions like the “Concours Lépine”. Thanks to those experiences, I know how to work with a team and to present project but this experience is another opportunity to improve those skills.

This project will also allow me to develop better coding skills, to learn how to do 3D modeling and to use different kind of software. In addition to that, everything that I will learn during this project is going to be useful for what I might do later as an engineer. That is why I look forward to learn how to create a video game with my team because I know we can learn a lot from each other.

Emeline Tichit:

I discovered computer science in high school through a programming class called “Informatique et création numérique”. We were taught how to code on Processing. Then, at the end of my senior year, our maths teacher made us try our hand at coding in Python. I therefore had very little experience in programming before joining EPITA.

I expect, through this project, to learn how to use Unity and Blender, gain experience in teamwork surrounding a game, as well as learn more about all the different steps related to the making of a video game.

Vincent Thirouin:

Ever since in high school I started developing games and applications and thus have a solid experience in software like Unity and Visual Studio but also with languages such as C#, Python and Java.

So, I was very excited to start working on the project and I think I can help my team by sharing my knowledge and my passion. I am someone very passionate and I tend to work very fast if the subject interest me. As we decided on this game idea together I am very excited to start working on this project with my team members.

But although I have been coding for a long time, I have never participated in a group project that big. This experience will teach me how to work in a group and will also push me to finish a game in its entirety from programming to graphics.

Being a lazy person this will also force me to work and to do my best in a limited amount of time. As I am not working alone anymore i will have to finish my part of the project before the deadline (and not 1 day before) to ensure that my teammates can work properly without worrying about me.

2 Presentation of the game

2.1 Bartendu

2.1.1 Concept

The goal of the game is to get as many points as possible in a limited amount of time. In order to earn points, players will have to successfully meet the needs of their customers. Those will usually be cocktails and drinks, but at times customers may also ask for sandwiches like hamburgers. There are many different steps that have to be completed:

1. Interact with customers who wait in line, outside of the bar, to make them come in.
2. Once the customers are sat down, their order will appear in a predefined space on the screen.
3. Players will then have to prepare the order, from picking the right container (glass or plate), to collecting and processing the ingredients (by cutting them in pieces for example), and finally adding them together to make the drink or dish.
4. When an order is ready, players will have to bring it to the customer. If the order is wrong, points will be taken off the global score of the round!
5. When a customer leaves, players will have to remove the dirty dishes from the table and wash them to be able to reuse them. Note that there is always a limited number of plates and glasses. This implies that players will need to be organized and cooperate with one another to succeed.

Moreover, during their games, players may have to deal with disruptive elements such as having to extinguish the fire that is burning the stove, or chasing the rats that may try to steal unsupervised ingredients or orders.

While this applies to the multiplayer mode, we also plan to build a tutorial mode and a singleplayer mode which will follow the same rules, with some variations specific to each mode.

The role of the tutorial is to teach new players the controls and how to play the game. Step by step objectives will be given to the players so that by the end of the tutorial they will have explored all the aspects of the gameplay.

Finally, a singleplayer mode centered around a story will be available. Even though we call it "singleplayer" mode, it will still be possible for multiple players to game together – we call it as such because it is not a requirement, players will be able to play on their own, as opposed to the multiplayer mode.

2.1.2 Levels

Levels will be designed to ensure that the game does not become redundant so that players are always entertained. Their layout will be adapted to increase the difficulty through the stages, and many elements can be implemented to surprise the players. Those include, for example, slippery grease stains or even ice floors, which will make it harder to control the movements of the characters. The goal is to force players to cooperate and think of a strategy, while also making it fun to do so.

2.1.3 Game design

The style of the game intends to be cartoon-ish, and relying on 3D models. A third-person view will be chosen, with a camera angle that will move according to the room the player is in. Keyboards will be used to control the characters' actions: cutting items, interacting with customers, using objects, holding and letting go of them...

2.2 (Cooking) Simulation games

2.2.1 History and definition of the genre

The game *Bartendu* is before anything else a simulation game.

An old genre: The genre of simulation is not new to the video-game industry. Indeed, it appeared first in 1963 with the game *Intopia*. It was a business simulation game made by the University of Chicago for their students to help them learn how to manage a company. This is the main point of simulation games which are by default trying to reproduce/imitate reality to make of a video-game experience something more real and useful for the players' day-to-day life.

Exploring many possibilities: The simulation genre had time to grow, expand and get mature over the last decades with its subdomains. Indeed there is not A simulation game genre but many. Over the years, they touched transport (e.g.: *Transport Fever 1&2*), sports (*Wii sports*), space, flight simulation (*Microsoft flight simulator* known as MFS), construction, racing, cooking (*Overcooked 1&2*) and more... It is quite diverse with many subdomains and gives many gameplay possibilities to players. It can be close to real-life experience or an approximation of real-life experience with added features and details that may vary from reality.

The cooking simulation genre is nothing new to the video-game industry. Though, it offered over the years many gameplay possibilities. To have a better overview of this domain, we'll use a few examples:

BurgerTime (1982): It is the first game of its genre. It was released in the United States first on Arcade consoles and Atari. It introduced the first basic concept: getting over a few obstacles to get a recipe done and serve the customer. However, a few recipes were implemented by developers in the game because of technical limitations.

The Cooking Mama series (2006 - ?): This set of video games on Nintendo consoles is a perfect example of the cooking simulation game taking a step closer to the real world. Indeed, with Nintendo DS's touchscreen, the game offers gameplay around slicing items, breaking the eggs and more. Finally, we can say that it added diversity with **many recipes implemented** and objectives to fulfill (serving customers with an objective of score).

Cook, Serve, Delicious! (2012): This game introduced nothing very new to the simulation game genre but added the business and **management** aspect. Indeed, satisfying customers added to many other new aspects such as recipe selection, restaurant upgrades. So, this game added the management aspect to the genre of cooking simulation. The goal being to be able to build a 3-stars restaurant.

Overcooked 1&2 (2017): The game is our main source of inspiration. It is the first to combine most of the aspects mentioned above. Indeed, it **combines** the constraint of time, customer satisfaction, and diversity of recipe to dynamic gameplay in **cooperation**. It includes many stages with plenty of obstacles for the players to win a game, but misses many of the management aspects mentioned with *Cook, Serve, Delicious!*

2.2.2 Point of this type of game

The main interest of this kind of game is for the player to be able to reproduce real-world cooking with the addition of management features in some games. Therefore, this fits better in the simulation genre. However, the ones playing the constraint of time or obstacles to avoid, add more diversity to the gameplay but simplifies the aspects of cooking. Nevertheless, they still stand in this genre but from farther.

Bartendu aims to be closer to *Overcooked* but wants to add some missing aspects to the game such as drinks recipes, management tools, and progression systems while keeping the main part of the fun available via the timer constraint. It will also feature the serving of meals at the restaurant's room as an additional gameplay possibility.

3 Production

3.1 Tasks distribution

Task	Main	Substitute
UI graphics	Lou	Emeline
UI implementation	Vincent	Lou
Character actions	Vincent	Jean
Multiplayer	Vincent	Lou
AI	Vincent	Jean
Tutorial	Emeline	Vincent
Level design	Emeline	Jean
Level implementation	Jean/Vincent	-
Object implementation	Emeline	Vincent
Particle Effects	Emeline	Vincent
3D modeling	Lou	Emeline
3D animation	Emeline	Lou
Music	Jean	Lou
Sound Effects	Emeline	Jean
Writing (Storyline)	Jean	Emeline
Website	Jean	Emeline
L ^A T _E X	Emeline	-

3.2 Description of the tasks

- **UI graphics:** Making sprites to make the menus pleasing to the eye is important, as it is the first thing they will see when launching the game.
- **UI implementation:** UI stands for user interface. It covers main menu, pause menu and in-game UI (order list, dialog texts and every text, sprite, button, and inputField). Overall, UI implementation is basically the making of everything the player can interact with and implementing it through Unity.
- **Character actions and movements:** We chose to control the player with physics and not just vector position translation. We apply a force to the player to make it move rather than changing its coordinates (x,y,z). We picked this implementation over the other because collisions will be frequent in our game, so we needed a way to avoid jerkiness and bugs. (Physics based movement cannot teleport through a collider because even if we apply a very strong force the block will counter us but with vector 3 translations we can just clip through)
Another good use of physics based movement is that it is more fluid and easier to synchronize with the other players.
Finally 4 actions will be implemented:
 - Throw, which applies an upward and forward force to an object
 - Pick and drop
 - Use
 - Boost (forward force applied to the player)
- **Multiplayer:** We first need to create a lobby : an interface for users to connect to different rooms. This requires a Master Client (someone that will host the game) and Clients that will receive the game information sent by the Master Client through

the server. We also have to deal with sending all the data via the server. Finally we will add player customization through custom properties to be able to give each player a nickname and much more.

- **AI:** We want to implement an AI that is playable both with and against. Our ultimate approach will be to train the AI using machine learning and an asset pack called ml-agents. But for now we will focus on implementing an AI using states
 - pick an object (path finding algorithm)
 - transform the object (path finding algorithm to the usable object (like cutting board) and use the object (cut the tomato))
 - brain (look at a recipe and deduce what items are needed)
 - logic (if an item is already crafted, do not redo it; if an order is going to expire, that one becomes a priority; learn to avoid map disruptions (thread mills))And other methods to make the AI intelligent enough. Then if we have time we would like to train the AI in an environment using reinforcement learning. The goal of machine learning is to implement actions that we could not have thought of, just by putting the AI in the game and letting it try to do its best.
- **Tutorial:** We want to help new players when they play for the first time. Therefore we will create a step-by-step scripted level that guides them through the different situations they will encounter in future games.
- **Level design:** We need to think of and draw maps that are playable and have different levels of difficulty; the layout of the levels will play an important part in this. Level design is decisive to make a good game, as players need to be entertained at all times.
- **Level implementation:** After creating the levels we will need to implement them through Unity.
- **Object implementation:** We will have to associate to each game asset its 3D model, some characteristics and properties. Indeed, items properties change when they are modified or used by the players.
- **Particle Effects:** Particle effects are nice details that one can add for a more enjoyable experience. They will be made for instances like fire, water running and extinguishers, to name a few.
- **3D modeling:** We will design all the objects (tomato, cutting table, working table, wall...) that we may need to use for the game.
- **3D animation:** Characters when they are in motion and some objects when they are used behave a certain way. We will have to articulate some of the 3D models to allow them to move properly.
- **Music:** Music is a huge task in this project because we aim to provide at least two original compositions to fit with the menu and in-game ambiance. It will require first the creation of sheets written by the team (mainly Jean on this task) and then a work with music software to produce a good-sounding music that feels like professional work.
- **Sound Effects:** Create original sound effects that will be associated to items and objects to make the game more lively.

- **Writing (Story-line)**: Pretty self-explanatory, writing the story provided by the singleplayer mode.
- **Website**: We aim to build a simple website using bootstrap css platform. It allows a flexible building of our webpages. Furthermore, GitHub pages provides a reliable hosting solution for the project's website. We will include on it pages concerning features, schedule, the team, and a download page with all our documents. The development will be quite fast because of the use of bootstrap's default theme.
- **L^AT_EX**: Write the Book of Specifications and the reports, in the most efficient manner possible. They have to be and look pleasant.

3.3 Schedule

Task	by First Defense	by Second Defense
UI graphics	5%	50%
UI implementation	10%	60%
Character actions	90%	90%
Multiplayer	90%	90%
AI	10%	70%
Tutorial	0%	0%
Level design	15%	60%
Level implementation	10%	40%
Object implementation	5%	45%
Particle Effects	0%	10%
3D modeling	15%	60%
3D animation	0%	50%
Music	10%	50%
Sound Effects	0%	15%
Writing (Story-line)	0%	10%
Website	80%	90%

We thought it was not necessary to add a column for the last defense, since it seems obvious that everything will be done at 100% for the final presentation!

3.4 Tools and resources

We will be using various tools for this project. The following is a potentially non-exhaustive list of what we will need to complete the tasks presented previously:

- **Unity3D** as a game engine, along with **Unity Teams** to facilitate teamwork and collaboration. Version chosen : 2019.3.0f5
- **Rider** as integrated development environment (IDE) for C# programming.
- **Photon** and more specifically the asset Photon Pun 2 to help develop the multiplayer environment.
- **Blender** to make 3D models for basically everything in the game: from the furniture of the maps to the items and characters. We will also use it to animate the characters and some pieces of furniture. We might also look into Sketchup, nonetheless Blender will remain the main tool for modeling.
- **Illustrator** for logos.
- **SFX** and **Audacity** for sound effects.
- **musescore3** to make the sheet music. To make the Computer music, Ableton 10 lite and cubase 10.5 are highly considered.
- **L^AT_EX** to write reports and their formatting.
- **Trello** for task managing and organization.
- **Discord**, **Whatsapp** and **Sharepoint** to communicate and share documents.
- **GitHub** and **Git** for a backup repository containing previous saves.
- **Bootstrap** to build the website and **Visual Studio Code** for the HTML and CSS part.

Conclusion

To conclude this book of specifications we can sum up and give an overview of the project. Bartendu is an ambitious project because it requires the use of many skills in various domains. Although Bartendu can be classified as a classic cooking simulation game, we saw that it is way more than just that because of the features we aim to implement. To do so, we have detailed a schedule and a task distribution that will allow significant improvements between defenses. And for this, the group is motivated and ready to push themselves to their limits. We know that Bartendu won't be an easy game to make but we carefully chose our features intending to make this concept a reality. This team is very strong as we all are passionate about that project. Furthermore, we each are experienced in different domains and have a lot to learn from one another. At the end of the day we are just a group of friends trying to have a good time while learning and practicing as well as discovering the work of an engineer.

Sources

<https://en.wikipedia.org/wiki/Overcooked>
https://en.wikipedia.org/wiki/Category:Cooking_video_games
[https://en.wikipedia.org/wiki/EpicMealTime_\(video_game\)](https://en.wikipedia.org/wiki/EpicMealTime_(video_game))
https://en.wikipedia.org/wiki/Hell%27s_Kitchen:_The_Game
https://en.wikipedia.org/wiki/Cook,_Serve,_Delicious!_2
https://en.wikipedia.org/wiki/Cook,_Serve,_Delicious!_3
https://en.wikipedia.org/wiki/Cook,_Serve,_Delicious!
<https://en.wikipedia.org/wiki/BurgerTime>
<https://en.wikipedia.org/wiki/Intopia>
https://en.wikipedia.org/wiki/Simulation_video_game