

# ULTIMATE FIGHTING CHICKENSHIP

## Report 1



TEMPÊTE GROUP  
March 2019

[charlie.brosse@epita.fr](mailto:charlie.brosse@epita.fr)  
[philippe.lefebvre@epita.fr](mailto:philippe.lefebvre@epita.fr)  
[david.horozian@epita.fr](mailto:david.horozian@epita.fr)

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>General comments</b>	<b>2</b>
<b>3</b>	<b>Game Mechanics</b>	<b>4</b>
3.1	Comments . . . . .	4
3.2	Movement . . . . .	4
3.3	Basic Attacks . . . . .	5
<b>4</b>	<b>Sound Design</b>	<b>7</b>
4.1	Music . . . . .	7
<b>5</b>	<b>Graphics</b>	<b>8</b>
5.1	Comments . . . . .	8
5.2	Map Design . . . . .	8
5.3	User Interface . . . . .	8
<b>6</b>	<b>Other</b>	<b>9</b>
6.1	Comments . . . . .	9
6.2	Multiplayer . . . . .	9
6.3	Website . . . . .	9
<b>7</b>	<b>Conclusion</b>	<b>11</b>
<b>8</b>	<b>Appendix</b>	<b>12</b>

# 1 Introduction

By the following document, the TEMPÊTE team presents to you the advancement of its project Ultimate Fighting Chickenship. The team has suffered a great loss during the first months of 2019 : indeed it lost one of its members, leaving the group with only 3 contributors. However, the group didn't give up : this document will be the proof.

This report will debrief by order of precision the advancement of each task presented in the Book of Specifications. We will start each section by general comments about how each task has been approached and realized.

# 2 General comments

After a few weeks of documentation, the TEMPÊTE team started the project UFC on the Unity (version 3.6.f1) video game design software.

As expected, we took a significant amount of time to fully understand the functioning of the major features of the software and the importance of each task that has been assigned.

Some features have been cut since a team member has been lost. As such, a character has been removed, as well as its dedicated map (Dead Poule), and some of the group's expectations have not been met.

Below is a table comparing our expectations (as written in the revised version of the Book of Specifications) to the actual results for this presentation.

	<b>Expectation</b>	<b>Reality</b>
<b>Game mechanics</b>	40%	40%
Movement	70%	70%
Basic attacks	25%	25%
Combos	0%	0%
<b>Sound design</b>	20%	20%
Music	30%	40%
Sound effects	0%	0%
Voices	0%	0%
<b>Graphics</b>	20%	10%
Map design	50%	30%
Character design	15%	0%
3D Animations	15%	0%
User Interface	0%	10%
<b>Other</b>	40%	60%
Multiplayer	50%	90%
Website	30%	70%
Artificial Intelligence	0%	0%

## 3 Game Mechanics

### 3.1 Comments

Implementing gameplay into the project was our first contact with Unity and, as a result, took a long time to be operational. We had difficulties understanding how C# programming changed from our Programming's projects and this one.

Indeed, programming movements and basic attacks were not instinctive at all and really different from what we did before. This is mostly due to the significant amount of features Unity provides the user to write scripts or to configure the scenes.

At the end, with much documentations, tutorials and patience (because of the bug resolving), we achieved what we wanted to do for this deadline.

A player prefab has been created to take in every modifications done on the player features.

### 3.2 Movement

At this point, a Ultimate Fighting Chickenship player is able to realize two different movements (see images): jumping (Space bar) and moving left or right (Q/D keys).

Screenshots for different movement abilities can be found in figures 1 to 4.

In order to implement movements, Charlie created a C# script called Player Controller. This script took care of the physics laws applying on our players : gravity and drag. Here the players are represented by two cubes : a small one representing the head and a bigger one for the torso. By using Unity Rigidbody and changing its position according to the inputs of the user : the player can move pressing the specified keys.

For the jump movement, we also implemented a boolean variable checking if the user can move or not (regarding its position with the ground) : if it is on the ground it can jump. This is to prevent the user from flying.

With a prefab, a problem has occurred : start positions or spawn points. Indeed, both players started the game at the same position. This is why we added two network start positions : the host will spawn on the first one and the client on the second one.

For the next, deadline we aim to enable player to realize a double jump : one will be able to jump once in the air. Also, we want the gameplay to be dynamic, as a result, we want the players to face each other no matter what happens in the game. For instance if the right player jumps over the left player and finds itself on the left on the map, they should be able to turn around and face each other again.

### 3.3 Basic Attacks

Charlie took care of the implementation of the first basic attacks : a punch and a kick. The method they used is : an array of BoxColliders (boxes able to detect collision) has been added to the player prefab as well as a Fighter C# script.

Each of those boxes represent a part of the body able to hit the opponent. The Fighter script contains a unique method LaunchHit taking into parameter the part of the body involved in the attack. The method detects every collision (OverlapBox method) with this part of the body and calls a function to make the other collider take damage. One can only attack by touching the Torso or the Head (multiplying by 1.5 the damages), the "Attack Boxes" do not act like "Hit Boxes" which means they can not take damage for the whole body.

For instance, at this point we only have a punch and kick dealing different amount of damage. If one 'punches' one's opponent : if it hits the head or the torso, the opponent will take damage.

Of course, the inputs have been updates in the Player Controller script : by pressing the specified keys, one can punch or kick. For the next deadline we would like to add more attacks, special abilities and a system of recovery time between attacks.

As a result, a Health C# script has also been added to the prefab. One is able to take damage and a system of health points proper to each character has been implemented.

## 4 Sound Design

### 4.1 Music

Philippe has the main role in sound design, and particularly in the music production. He used his previous experiences in music creation to build the main theme of Ultimate Fighting Chickenship. Along with his musical genius, he used Ableton (see fig. 5) to build the music that can now be heard in the main menu.

The music is essentially 1min20 loop that repeats itself. 1min20 is enough time to have interesting variations but short enough to ensure a small music file.

The standout element in the song is the chicken shouts repeating themselves in a rhythmic fashion. Along with these shouts are epic percussions with fast strings to add rhythm. He added fast-paced Hi-hats to bring out a hip-hop vibe to the song. A flute can also be heard. In the second part of the loop, the epic percussions switch with trap drums, which helps break the monotony of the loop and enhance the hip-hop vibe we wanted the main song to also have.

With a main theme done and a musical direction that is now fully defined, the amount of work needed to produce the rest of the game's music will only be facilitated.



## 5 Graphics

### 5.1 Comments

Graphics is one of the main points that was completely shaken due to the loss of a team member. We had to reduce the goals we had set, in character design / modelling as well as map design for the first deadline. For this reason, the focus of the team has shifted, leaving only the basic graphics completed.

### 5.2 Map Design

David has started work on Krispoul's map: a replica of the background of Algorithmics' Mimo videos has been created in Adobe Photoshop and is used as the background of the (currently) single map of the game. (see fig. 6)

Resources that were used to recreate the background include:

1. [https://www.flaticon.com/free-icon/python-file-symbol\\_28884](https://www.flaticon.com/free-icon/python-file-symbol_28884)
2. [https://www.flaticon.com/free-icon/html-file-with-code-symbol\\_29515](https://www.flaticon.com/free-icon/html-file-with-code-symbol_29515)

For the next presentation, we aim to have at least one other map to present, if not all three.

### 5.3 User Interface

David has also created a basic main menu in which the player can choose to start the game or exit the game. In the main menu, a loop of the main theme is played as long as the player doesn't click on a button. (see fig. 7)

## 6 Other

### 6.1 Comments

Our advancement of the multiplayer integration is really satisfying because we have reached our goal. An early implementation of the multiplayer took us quite a bit of time early on, but thanks to that, the amount of work will be reduced in the future.

### 6.2 Multiplayer

Philippe has done the research regarding the multiplayer implementation. We have determined that implementing it from the start of the project is the best solution regarding the time it would have took to convert a single player game to a multiplayer one.

Our game scene is containing a GameManager Prefab that handles the networking abilities of the game. This prefab contains the Network manager component as well as the Network Manager HUD component. Network manager component handles the multiplayer and HUD component allows us to have a premade menu with basic features such as LAN Host and LAN Client.

Then each object in the scene that has a server-related compartment must have a Network Identity Component, and a Network Transform component.

- Network Identity will define if the object is server based or if the client will have authority on the object
- Network transform synchronize the object with the server. It has parameters for the frequency at which the information is sent or what the movement threshold is.

Before doing any action, the object must verify if the client has authority over it. This way, the player can only move his character.

### 6.3 Website

David began working on the website as soon as the project started. Using his previous knowledge of HTML and CSS, he put together a simple homepage that was updated as the project progressed.

The website began as a simple set of empty pages, that was transformed into a real layout:

- *Home*, a few paragraphs to introduce the website, the project and the team,
- *Project*, a page where we post advancements, screenshots and other various news for our community to see,
- *Download*, a page that will host download links to the game and various reports as they are completed and ready to be published,
- *Links*, a page that will host all information and media that was used to create the game, such as sources for images, designs, models and so on.

As of the 1<sup>st</sup> presentation, only the first two pages (Home and Project) are available (see figures 8 and 9)

Resources that were used to create the website include:

- *Brad S Knutson's CSS animations*, for the animations in the navigation bar **(1.)**
- *w3schools' HTML and CSS references* **(2. and 3.)**
- *Google Fonts*, for the beautiful fonts that were used in the website: Source Code Pro, Roboto, Lato and Muli **(4.)**
- *Atom*, an open-source editor with features such as auto-completion of HTML and CSS tags, a very useful tool and time saver **(5.)**
- *GitHub Pages*, where our website is hosted **(6.)**

Link to resources:

1. <https://bradsknutson.com/blog/css-sliding-underline/>
2. <https://www.w3schools.com/tags/default.asp>
3. <https://www.w3schools.com/cssref/default.asp>
4. <https://fonts.google.com/>
5. <https://atom.io/>
6. <https://pages.github.com/>

## 7 Conclusion

In conclusion, the game has advanced to a satisfactory point, even though we have had to change our plans to adapt to the loss of a team member.

Game mechanics have advanced at a rate that respects our previous schedule. Music has exceeded our expectations, with a theme already created and only some ambient sounds and tracks left to be added. The website is already close to being complete and the multiplayer is already very advanced, leaving us a good place to start focusing on adding features.

On the other hand, graphics was a domain that was in majority attributed to Maxime, and with his absence we have had to double our efforts to have at least some bases to rely on. Finally, artificial intelligence was not a point of focus for this presentation.

## 8 Appendix

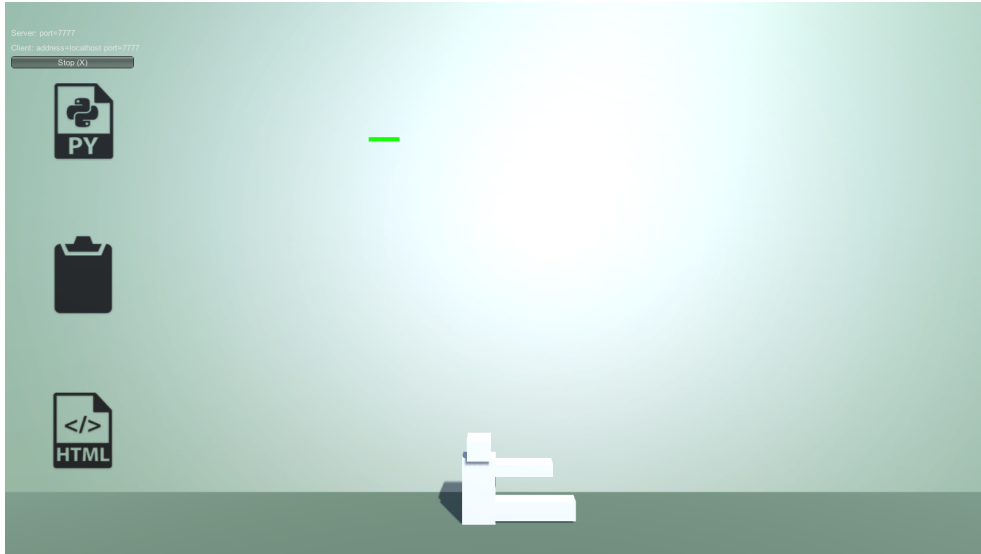


Figure 1: Normal position



Figure 2: Moving left

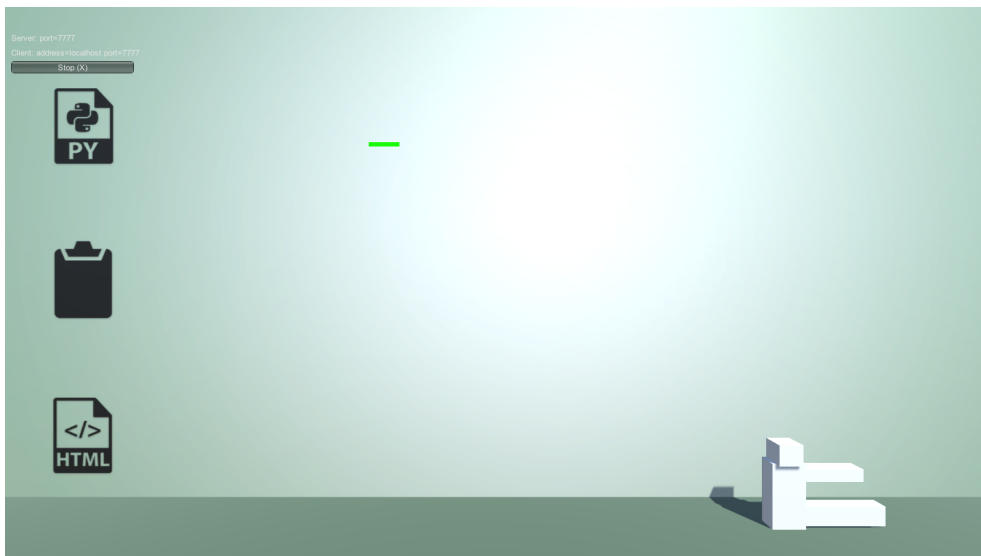


Figure 3: Moving right

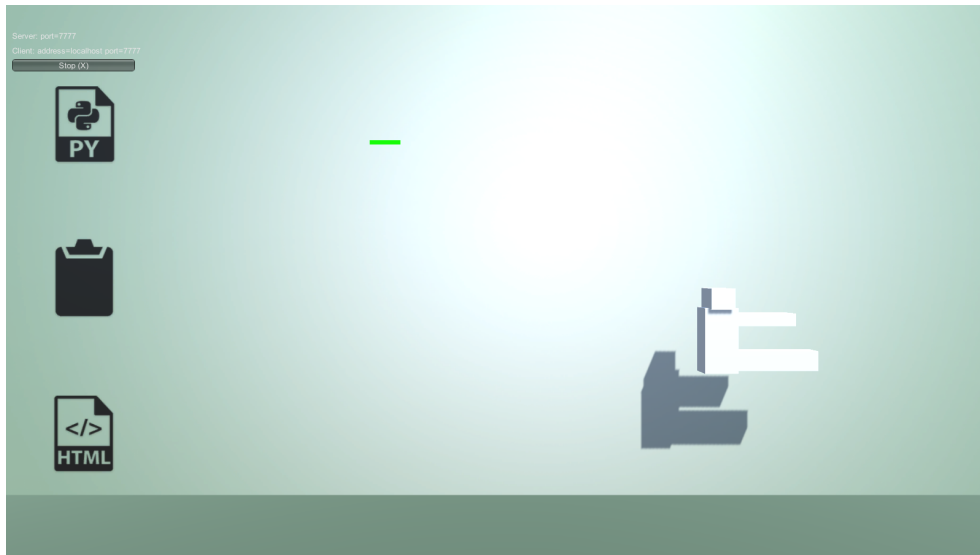


Figure 4: Jumping

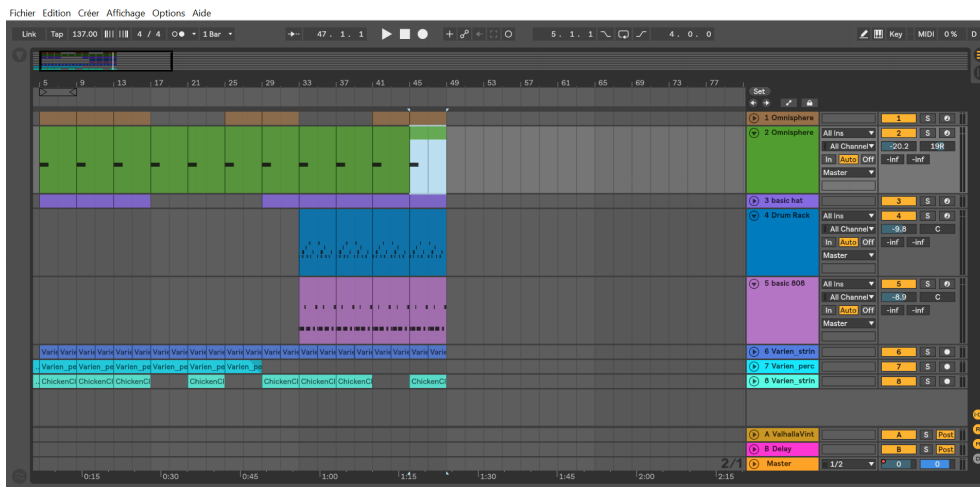


Figure 5: The creation of the soundtrack in Ableton

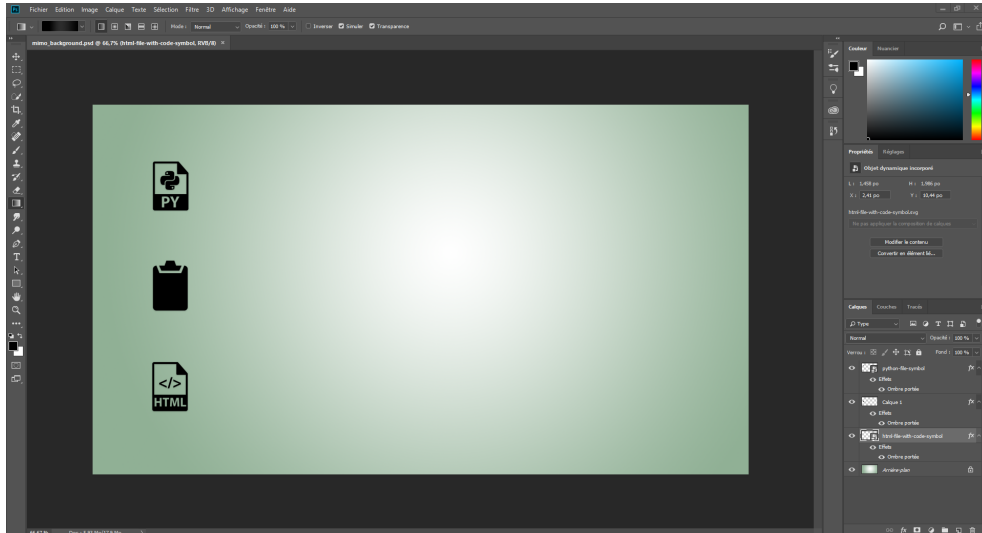


Figure 6: The creation of the Mimo background in Photoshop

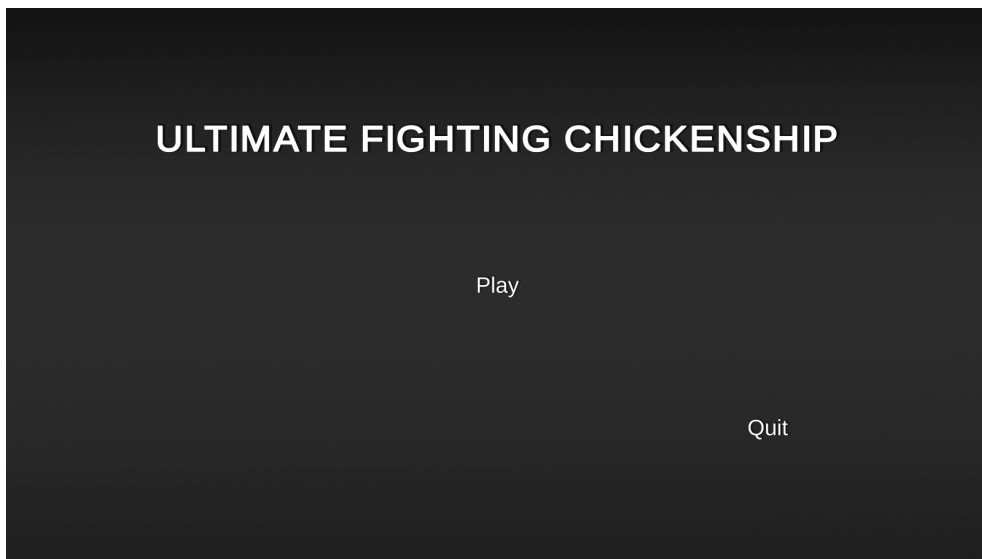


Figure 7: A screenshot of the main menu



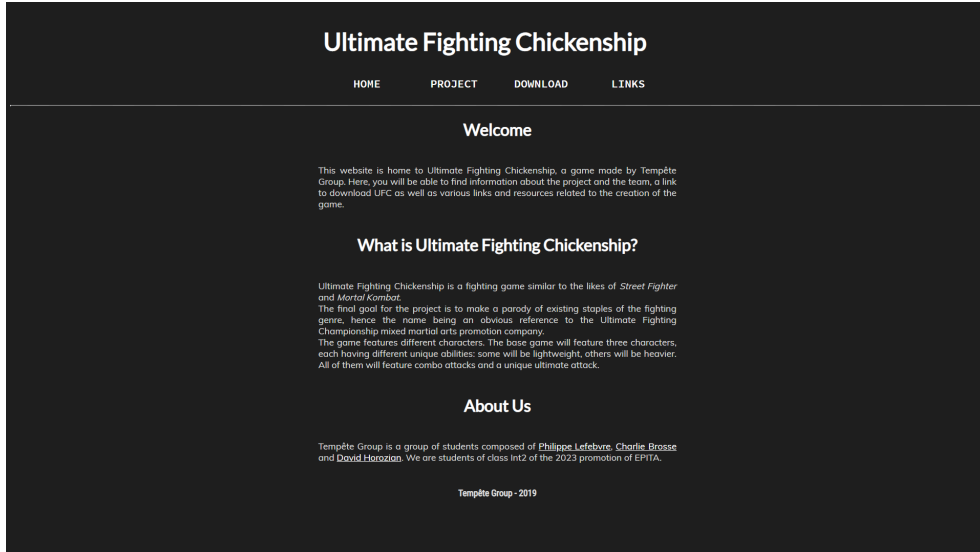


Figure 8: A screenshot of the homepage as of March 2019

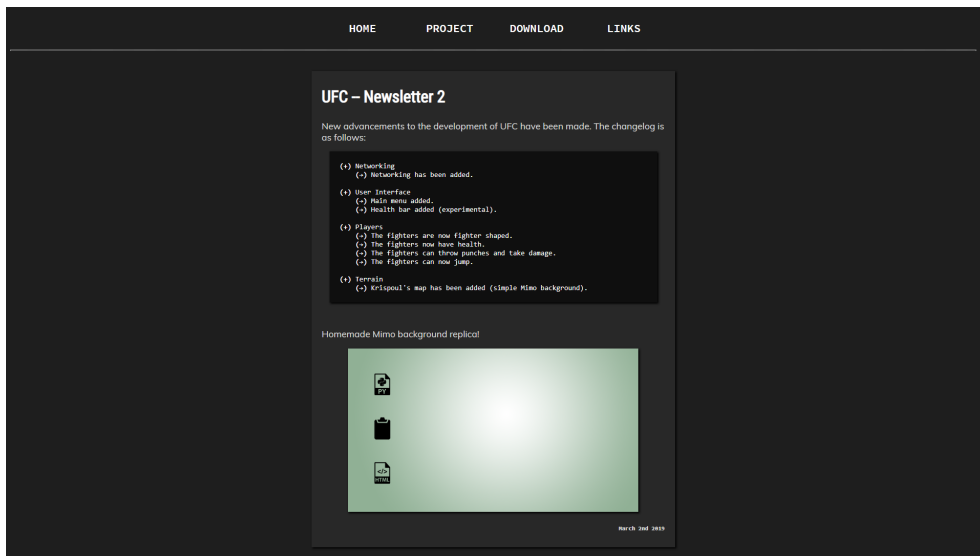


Figure 9: A screenshot of the project page as of March 2019