
PRACTICAL 3

130023531

INTRODUCTION

In this practical we were asked to develop and test a game in the processing environment. This includes providing a description of the game design and the decisions taken while developing the game.

GAME DESIGN

TITLE

The game is called Blobby, the name came from the blobs (food) in the game.

GENRE

This game can be classified as an online multiplayer game with some sort of action and crafting components to it.

PLAYERS

In my game, each player is represented as a cell. The players can control the cells using the mouse and they can craft and use abilities with the keyboard. In order to differentiate each player's cell, the screen will always be centered on the current players cell and users can have nicknames on their cells.

OPPONENTS

As for the opponents, in my implementation they are other players controlling cells over a network.

RULES/MECHANICS AND GAME COMPONENTS

In my game players are represented by cells which they can control using the mouse. The purpose of the game is to be the largest living cell in the world i.e. the first cell on the leaderboard. In order to do so, the player will have to eat food blobs or eat other smaller cells to grow. To make the game more equal, I have added abilities which the players can use so that size is not the only thing that gets into account when it comes to eating other cells. Players will be able to collect resources on the map to be able to use/craft abilities. Each player will have four abilities, three of those four will be the same for all players except the last one. The last ability or what I like to call the ultimate is unique to each cell type (champion) and they do not require any resources to run, however, they are on a time cooldown. The different abilities and different champions will make the game enjoyable for players and let them develop their own strategies and tactics.

PLAYER MOVEMENT

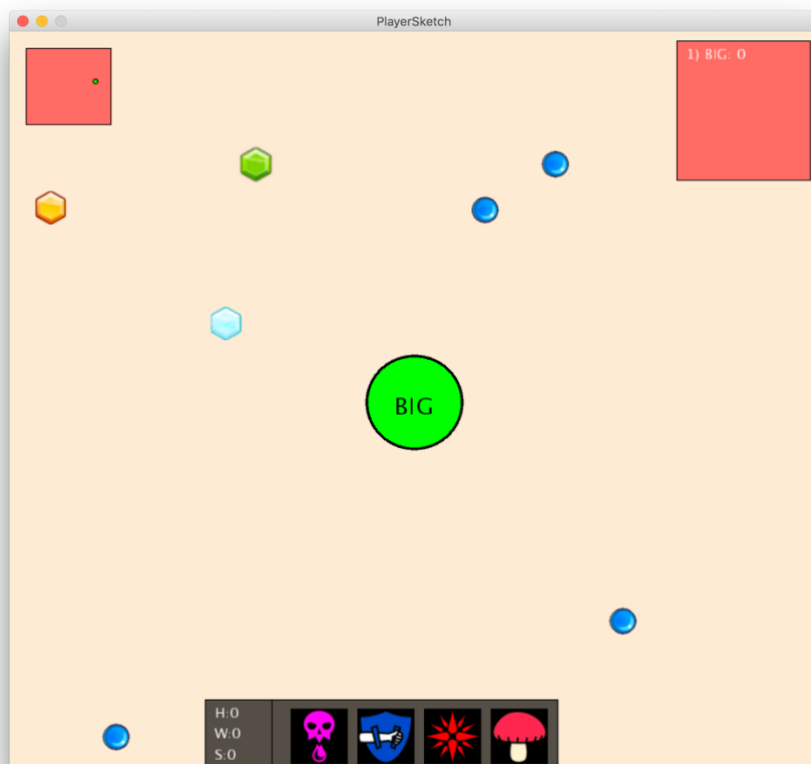
The player will be able to move the cell they are controlling by using the mouse. The cell will follow the direction of the mouse and if the player wants to keep the cell still then the mouse should be on the center of the cell.

As for the speed of the cell, then that depends on how many food blobs the player eats. The more the players eats the slower the player gets.

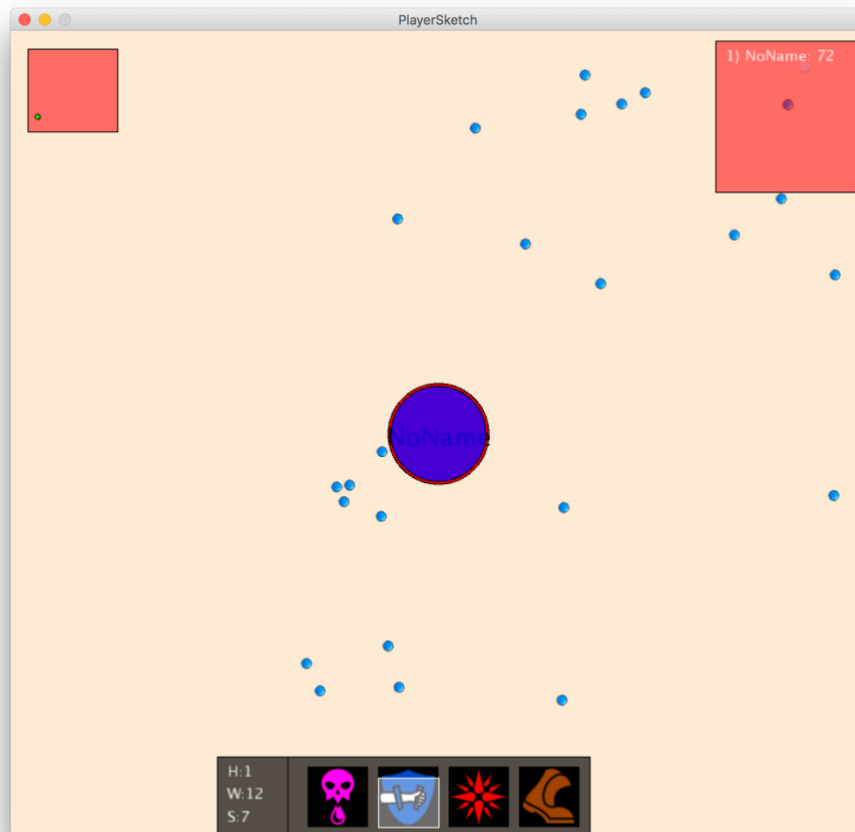
DISPLAY

Visual Aid:

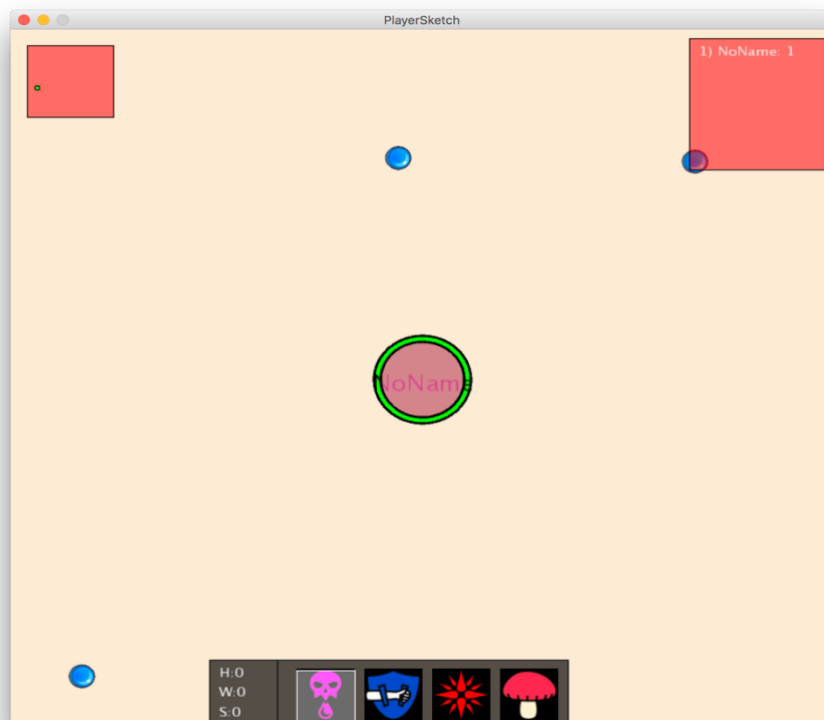
- **MINI-MAP, Inventory and Zoom Out:** In order to help the player navigate through the world or at least to know where he/she is with respect to the world I have implemented a **mini-map** which shows at the top left of the screen. The inventory at the bottom left shows how much of each material you have: H for Herb, W for Wood and S for Steel. In addition, the screen zooms out as the player gets bigger in order for the player to have a better look of potential dangers.



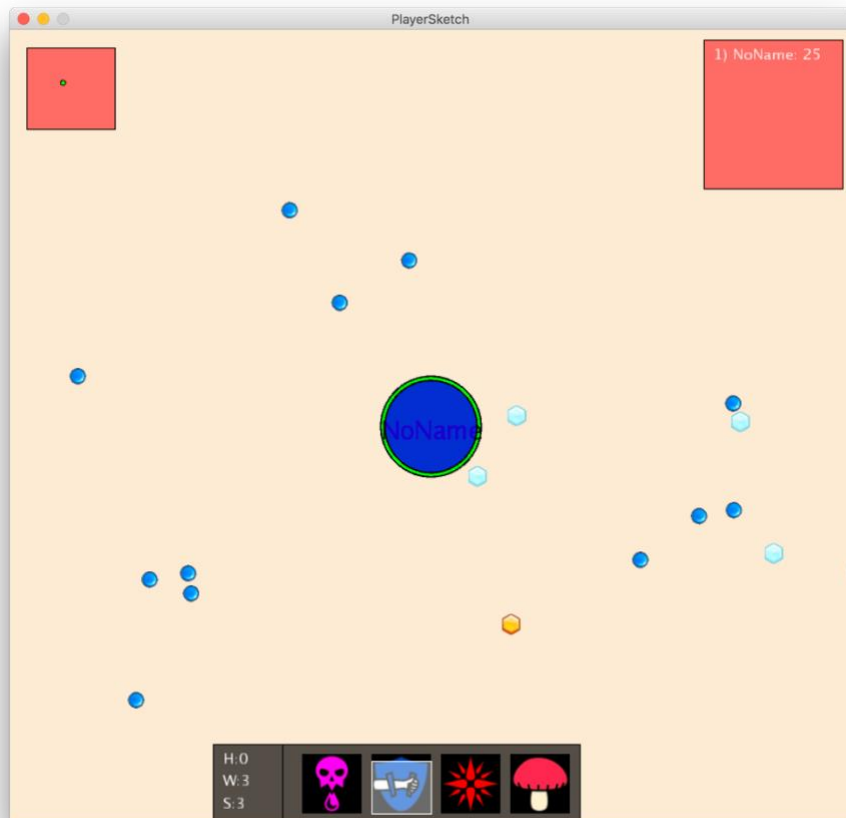
- **COOL-DOWN TIMER:** In order for the players to know how long left on their abilities, I have made the ability boxes slowly fill up to indicate how long is left. Once the box is fully filled then the ability effect ends. In the figure below this can be seen on the second ability box from the left.



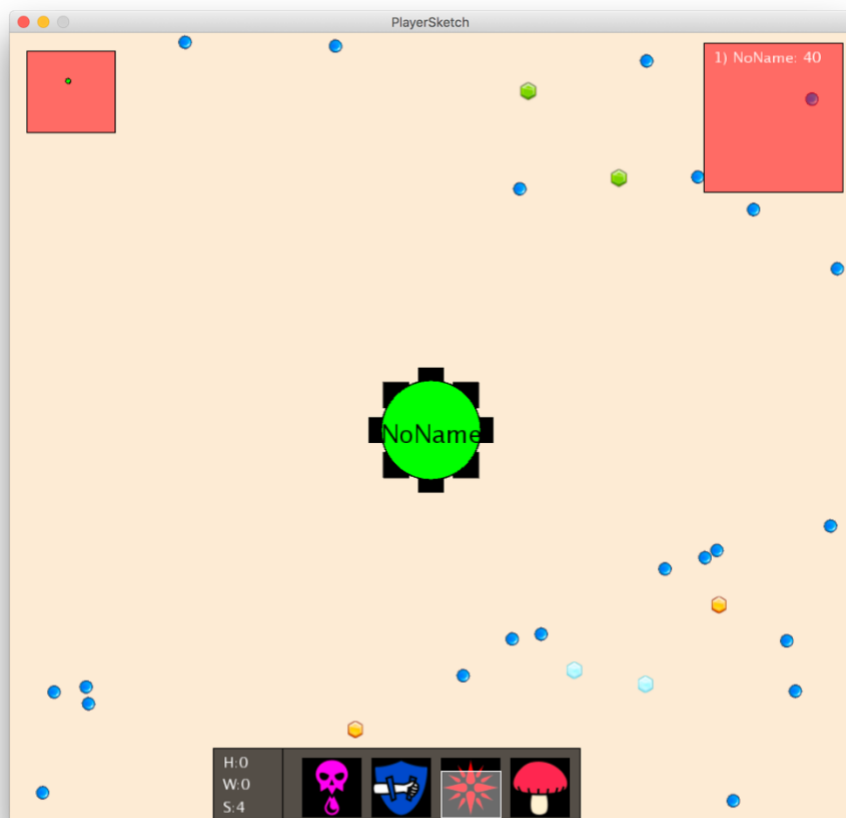
- **CELL STATUS:** In order for the player to know what ability to use next, it is important to see if the other cell is poisonous, if it is shielding or if it is spiked:
 - **Poisonous:** It gets covered by a pink layer.



- **Shielded:** It gets covered by a blue layer.



- **Spiked:** It gets surrounded by spikes.



- **Pick-Ups:** these include the materials that are used to craft abilities and the food that the player can eat to grow:

- Food:



- Herb:



- Wood:



- Steel:



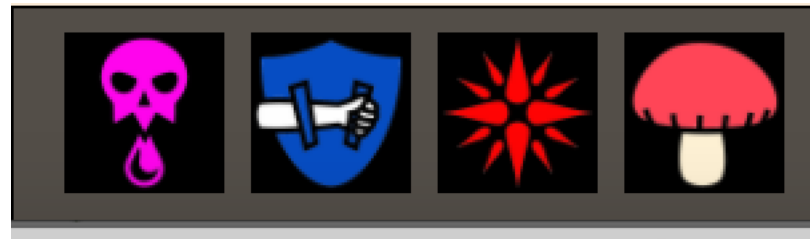
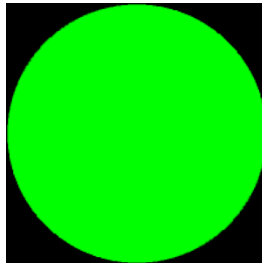
CELL TYPE

This game has four different type of cells. The player can choose between many based on his/her preference. All four champions can craft the same abilities, but they all have different ultimate. Below you can see a summary of the champions and their ultimate's:

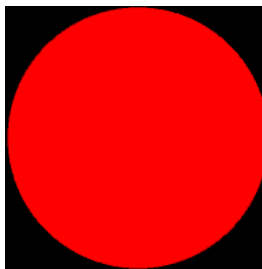
- **Riven:** is known for her poisonous mass sucking mushrooms (or as known in the game Shrooms). Riven's ultimate allows her to drop a Shroom at her center. If any unshielded cell comes in contact with the Shroom, then that cell will die and Riven will absorb the mass of the other cell. Riven's ultimate is on 2000 frames cooldown, which start from the Shroom drop.
- **Zed:** is known for his speed. Whenever Zed uses his ultimate he gets a boost in speed equal to twice of his current speed. The effect of Zed ultimate last for 50 frames and his ultimate is on a 1000 frames cooldown.
- **Lulu:** is known for her size. Lulu is able to increase her size by half of her current size and it lasts for 50 and her ultimate is on a 1500 cooldown.
- **Jax:** is known for his ability to generate enough resources to be able to use craft the three items once. His ultimate is on a 2000 frames cooldown.

Below I will show what is the color of each champion and how its inventory looks like:

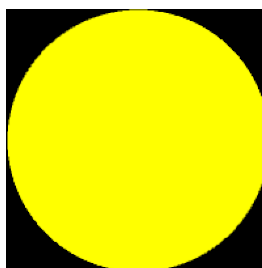
- Riven:



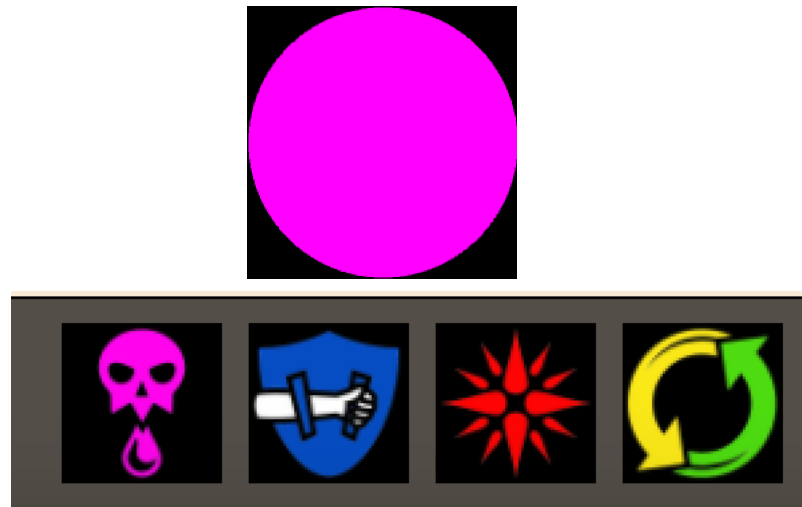
- Zed:



- Lulu:



- **Jax:**



PLAYER ACTIONS

In addition to moving the cell around the player is also able to perform some actions. Whether the player is capable of performing a specific action really depends the number of resources the player has (this gives the game some sort of crafting sensation).

In this game, we have three types of materials: Herb, Wood and Steel. These can be collected by the player to craft three items: poison, shield and spikes.

- **Poison:** In order for the players to craft poison they need to collect two Herbs. Whenever the two Herbs are available the player will be able to make poison by pressing the Q button. This will make the player poisonous for 50 frames. If any cell tries to eat the poisonous cell, then the poisonous cell will affect the other cell with poison, kill it and take its mass.
- **Shield:** As for the shield, the player needs to collect two Wood and one Steel. The player can activate the shield item whenever the resources are available by pressing the W button and it lasts for 50 frames. The shield will make the player immune to both poison and spikes. In case of poison, the player will remain immune to poison as long as the shield last the whole duration until the poisonous cell becomes not poisonous again. As for the spikes, whenever the spikes hit the shielded cell then the shielded cell will lose the shield and the spiked cell will lose the spike.
- **Spike:** For the spikes, the player needs to collect one Herb, three Wood and two Steel. The player can then activate the item pressing on the E button and it lasts for 50 frames. The spikes can be used as a gap closer between cells. Whenever, the spike touches another cell it will kill that cell right away. However, if the other cell is shielded, or the other cell is spiked then both cells will lose their respected item.

The last action can be activated with the R button. This controls the champions special ultimate which has been described earlier.

MECHANICS

When it comes to game mechanics, it basically depends on how much does the player practice. As the player plays more and more different strategies can be developed on the way. For example, for each champion there is a certain way to play with that champion. If you play Lulu for example then you should be able to focus on collecting resources for shields and then aiming to eat cells as your ultimate makes you twice as big. There are many strategies that can be developed some of which I can't think of, so mechanically it is an open-end game.

GOALS

The ultimate goal of my game is to reach the top of the leaderboard. My game has no ending and you cannot really set a winning point. All players will compete against each other in order to get the first position on the leaderboard. While competing for the first rank, players can set smaller goals for themselves such as chasing smaller cells in order to grow or trying to destroy other cells to get bigger.

CONTEXT

My game idea was based on a combination of Agar.io and MMORPG. The game is similar to Agar.io in a sense that there are cells and they can grow by devouring food blobs and other cells. However, in Agar.io it is usually hard for players that started small to reach and compete with players on the top rank. Thus, I decided that it would be better to have a way where players can use abilities and items to compete against each other. By collecting the appropriate resources, players can craft and use three items along with an ultimate which does not require resources but is on a time cooldown. The ultimate differs for each cell type (champion) just like MMORPG.

DESIGN

CODE STRUCTURE

In this implementation, I have divided my game into several packages:

- **GameObjects:** It contains all the objects necessary for the game such as the Cell, Item, Material, Shroom and Inventory.
- **GameTools:** It contains all the classes needed to control the game such as the GameLogic and the class needed to draw the game objects, GameDrawer.
- **Network:** It contains all the classes required to perform a network connection. It contains both the GameServer and the Client in addition to the PlayerAgent which holds information about the connected clients. It also contains the StateMessage which is what the server sends to the client and the PlayerMessage which is what the client sends to the server.
- **Players:** It contains the classes that represent different types of player, ClientPlayer and LocalPlayer.
- **Sketches:** This package contains all the executable processing sketches. Running ServerSketch will run the server, running the PlayerSketch will run the player from which you can then choose to either be a local player or a client player.

GAME DESIGN

I have decided to make this game a multiplayer game where players can compete against each other. To make the game more interesting I have decided to include resources which the player can collect in order to craft usable items.

The first thing I started working on was the cell movement, I noticed that the best way of moving the cell accurately was by using the mouse. Initially I tried moving the cell with the arrow keys but the movement felt unnatural. The next thing I looked at was the speed of the cell, I had to find an appropriate starting speed which I managed to figure after multiple runs and testing.

After figuring out the speed, I worked on generating the blobs (food). As the game was based on competition I decided that it is better that I randomly generate extra blobs whenever the total number is less than 200 blobs. This makes sure that we won't have players being preferred over others.

As for the materials, I decided that have a lot of material is not a good thing but not having any is not good as well. This is why I have made my game in a way such that there should be at least 10 of each material and only generate when the number goes below that. (it might seem a lot of resources for one player but if we were playing against more it would be more than enough).

After implementing the blobs and material, I moved along to work on the collision detection. As everything is a circle it is quite simple to implement the collision. In case of cell-cell collision, I decided that there should be an overlap of one cell over the other before one of them be able to eat the other. This was to make the game more challenging.

Once the collision detection was done, I moved to work on the inventory and item crafting. I found it better to have three items which all players can craft and one single ability which is unique to each player. In my opinion, this allows different players to try different champions based on their play style and they do not have to learn a lot of new stuff every time they switch champions. As for the item timers, I found that it makes more sense to leave all of the three items running for the same period of time. Due to the latter, I had to ensure that all of these three abilities will be equal in their life span. The way I ensured that is by adjusting the results of how the items will interact with each other when their cells collide. One example is the case when we have the shield and the spikes, if I let them work on the time base on of them will eventually run out before the other realizes what is going on. To overcome that, I made their collision result in both cells losing the shield and the spikes which ensures equality.

After I finished implementing the items, I moved to implement the ultimates. The champion names were inspired from a game I play a lot which is League of Legends (LOL). I tried to make the ultimate close to what each champion does in LOL while making sure that it will be useful for this scenario by tweaking it a bit.

In addition, I have incorporated a live ranking system that shows what is the current rank of the player this is done using the `Collection.sort` function with implementing the comparator. The ranking always shows the top five but it will also show your ranking. Other players will appear with a black nickname while you will appear with a white nickname. This can be seen in the figure below:



As for the networking part I decided to go with UDP over TCP. Even though that TCP ensures that the data packets are delivered to the clients, it is slow due to it maintaining the ordering of the packets and making sure that the packets actually are received by the receiver. However, UDP offers a faster data packets transfer as the sender will not wait to make sure that the receiver received the package. This is the protocol that I opted to use in my implementation as in the game the server will keep sending the game state each frame and even if something goes wrong the players will always catch with the server because they only draw the state sent from the server. This is how my networking works:

- Server starts running.
- Player sends a join request, Server replies with the player Id.
- Server sends the game state and the Player updates its current state with the one sent by the Server.
- Player sends back to Server mouse movements or key presses.
- Server applies the actions on the current game state and sends game state back to Player again.

As I initially intended to implement game + AI, I had a local game which later on I switched my game to be ran over the network. I kept the local game play, however, I do not have AI for players to play against. Thus, I decided that it would be a good idea to have the local play option as a practice area where players can see how the game progress, how speed differs as you get bigger and what is the best item combo.

To minimize the possibility of cheating I reverted from using a P2P connection and opted for a single authorized server were players connect to. This ensures that the states received from the server is always trustable and thus reduces tampering with the game.

I am really proud of my menu selection which you will be able to see in the submitted video. I am also proud of the whole networking bit of my game as this was the first time I implement my own network. I am also happy with how the game turned out with different champions and items, it made it more fun.

Given more time I would've worked on creating more champions and implementing a bot which helps players to practice in the local arena.

PLATFORM

This game was built to be played on a desktop.

EVALUATION

I tested my game by playing it myself multiple times and also by letting my colleagues try the game.

When I first started designing this game I did not consider implementing networking for it and I based my game on a local game play where the player will play against artificial intelligence. As I was developing and building up the game, at a point I had a basic random AI working where it searches for food and heads towards it. At that point, I let my colleagues try the game and most of them agreed that my game implementation would be better if it was over a network. I listened to what the recommended and I thought that it actually made sense as the game is more competitive than something you would play to relax. I had to re-structure my whole game implementation to include networking and this was one of the most difficult things I have done as the I have already implemented the game to be played locally. However, I managed to make it work and I decided to remove the AI.

Another comment which was provided by one my colleagues was the ability to know how long the item will be activated for. This pushed me to implement the timer on the item boxes.

CONCLUSION

This practical was interesting to work on as it showed me the process of designing, implementing and testing a whole video game.

REFERENCES

- Icons used from [<http://game-icons.net>]