Requirements for Multiplayer- S5

By group 26 - The Fishermen

In assignment 4 we were required to list the requirements for the TA exercise. At the meeting with the TA on 20/10/2015 we were told that we could finish this feature as the 20-time in assignment 4. These requirements are therefore not very new, but a few changes have been made. They are shown in italics. The most important thing we'll work on is fixing the client receiving settings and allowing the client to move.

The requirements are listed in the same way as our game, so we will use the MoSCoW method. Firstly, the functional requirements of the logging are listed as must, should, could and won't haves. Secondly, the non-functional requirements are listed.

Functional Requirements

1.1 Must haves

- The game Fish.io must have non-local multiplayer.
    - This must support multiplayer for at least 3 clients.
    - The connection will find place through TCP/IP connections.
- The implementation will be socket-based and will use a server that distributes the necessary information and environment variables.
- The following actions must be implemented:
    - Every client must be able to control its own playerfish.
    - Players should be able to eat another player if they are big enough according to the game rules.
    - Smaller AI fish must spawn and move so that the players can eat those instead if they cannot eat players.
- Multiplayer must be implemented into the GUI.
    - There must be a multiplayer screen on which the player can connect to a multiplayer game by connecting to a specific IP.
    - There must be a button in the main menu that links to the multiplayer screen.

1.2 Should haves

- *Multiplayer mode should have a background.*
- *The client should be able to differentiate different users by seeing them with a different color fish.*
    - *All player fish should still have the original player fish sprite.*
- The settings should be the same on all clients.

- ○ *The server will send a settings message to a client when that client connects.*
- The playing field should be bigger than a single player playing field..
- The user's view of the playing field should be able to move within the boundaries of the playing field.
  - ○ This is called a viewport and will also work for a single player game in case the screen is smaller than the field.

1.3 Could haves

- The server could spawn certain power ups that players can collect.
  - ○ These might have to be rebalanced in order to be fair to other players.
- There could be extra entities like a large fish that is able to eat all players.

1.4 Won't haves

- The multiplayer implemented won't be local multiplayer version (2 players on one machine).
  - ○ (The items below are requirements of local multiplayer):
  - ○ The user shall be able to start a local multiplayer game using a "multiplayer" button on the start menu.
  - ○ The local multiplayer version shall work similar to the single player version. A few differences:
    - ■ The controls for the extra player(s) will differ from the controls for the first player.
    - ■ Different player fish can have overlapping bounding boxes without any result.
- There won't be any extra entities like:
  - ○ Sea urchins: When the player is bigger than a sea urchin and collides with it, the player gets killed instead of eating it.
  - ○ Obstacles: the player must move around obstacles and cannot move through them.

Non-functional requirements

- The second, working, version of the multiplayer mode shall be ready and submitted to the staff of the course on the 23rd of October.
- The multiplayer will not be tested with the same coverage as the total game (75%), as this is very difficult to do right on travis and it would take too much time to implement.
- The components that can be tested easily on travis will be tested with the same coverage as the entire game (75%).
  - ○ These components will be identified as multiplayer is implemented, and will be noted in the UML.

- The multiplayer implementation will have to adhere to the same non-functional requirements of Fish.io itself, with exception of the modified non-functional requirements in this document.