Students: Cainarean Constantin and Denis Garabajiu

1.Problem Description and Analysis

From the start we were given a working single player asteroid game, our task was to make it multiplayer and to implement a data-base where all the scores would be stored. In order to be able to perform this we needed to start with **Networking.** For creating a multiplayer game, it is a better choice to use **UDP,**  so we started by creating the Server <-> player interaction. Our plan was that we would have a main server, where all the players will connect and communicate with each other via server. In order to do this, we would need to send packets to and from the server. The PacketModel is our general class for packets. Next we did the ServerModel and the Player. The assignment also said to add different functionality such as player nickname, different colors of the ship and many others. In order to be able to get the assignment done on time, we devided the work so Constantin worked on the Back-End and Denis on the Front-end. We discussed how our architecture will looked to us and began to work.

2.Design Description

As stated before, we have a main server, where all the players will connect to and will communicate to each other via packets. Server and Client extend Thread, in such way the implementation of the communication is more easy to made. In addition to this, Server is a GameUpdateListener, so it will send update packets to the Player every gametick.

Our whole project is based on the **MVC** pattern, we have more than 3 packages in our project, but the frames can be found in the view Package, Control in control and Model in model. We modified our game constructor to be able to implement Normal multiplayer mode, Spectator Mode and Duel Mode.

Join Mode: The Server starts, the player connects to the server and once in a game tickrate, the server sends update packets to players and players send their current state of the game to the server so it can register the changes on it’s local running game.

Spectate Mode: The server starts, the player connects, but he never sends Model packets to the server, only receives updates on the game.

Duel Mode: The server launches a new game, but without asteroids. Player does the same. Player connects to the server and the duel model starts.

3.Evaluation

We tested our program on Windows and MacOS, we didn’t check it on Ubuntu, but it should work because Ubuntu and MacOS are similar systems. Sometimes we get a bug where the game lags, we think that this happens because of the big number of players and asteroids on the server, so it cannot process all the information correctly.

4.How work was divided

Most of the time we worked together, starting from the beginning when we draw our architecture and finishing with the view, however Constantin worked more on UDP, database and Model, while Denis worked on View and Control. It was a pretty rough assignment, but we managed to finish it. We know that maybe we have some more bugs that we don’t know about, but we think that we created a decent game.

ARCHITECTURE BELOW:

