COP 290 - Assignment 3
Battlestar Galactica

Akshay Kumar Gupta
2013CS50275

Haroun Habeeb
2013CS10225

Barun Patra
2013CS10773

J. Shikhar Murty
2013EE10462

Abstract

The game is a real-time racing cum FPS game. It takes place in space. Each player owns a spaceship and the objective is to be the first to reach the end. Players can also shoot at other players and try and destroy their ships. Each spaceship has different stats like accuracy, speed, ammo etc. The game is designed for a maximum of four human and four AI players.

1 Overall Design

2 Network

We will use UDP to handle network connections. It is a completely connected peer-to-peer network.

2.1 Start of game

One user starts the game and the P2P network. This user specifies how many people will be joining the game. A new user can connect to the network by sending his IP to any existing player in the network. As UDP is being used there is no notion of an actual 'connection', all each user needs is the IP address of each other player. The existing player will give the IP of the new user to all other users and the IP of all other users to the new user.

2.2 Message Passing

- A user broadcasts his state to all other users at regular time intervals.
- We have a separate thread for listening to incoming messages.
- When a new message is received from another user it is passed on to the rendering thread which renders that user according to his state.

- In case another user sends us a message which drops, the new position of the user is simply extrapolated from his old position, velocity, and acceleration.

2.3 Handling Drops

If we have not received a message from a user for the last T seconds, then we assume that the user has dropped. T is large enough compared to the average time interval between messages for the probability to be very low that the user has not dropped.

3 Graphics

3.1 Model Rendering

- Blender will be used to create models of spaceships, asteroids, debris etc.
- We will load the object files for these models using our own code.
- We will also have our display function for each model.

3.2 User Interface / HUD

3.3 Camera Positions

4 Music

Music will be incorporated in our project using openAL.

5 Physics and Collision Detection

5.1 Collision Detection and Handling

5.2 Hit Detection

6 AI

7 Testing

8 Planned Improvements