CSC 481 Fall 11 Homework 3: Multiplayer Extensions  
Due: 11/11/11 at 3:50pm

Overview

Your task for this assignment is to explore networked multiplayer architectures. You will be using the Processing programming environment for this assignment. You will submit your code and a writeup of your results, including screenshots where appropriate. The main assignment is an individual assignment, you are to work alone. For the extra credit portion of the assignment, you must work with one other person to implement the code, but you are to do your writeup individually.

Part 1, Remote Visualization

For this part of the assignment you will be laying the foundations for creating networked versions of your game. Think of your current codebase as the server, and create a “remote viewer” that will run on another machine (or at the minimum, in another process communicating using the loopback interface). Hint: You should be able to leverage the remote logging code you created for Homework 2 by considering all object movement as an event. You should also be able to modify our existing engine code to act as the client. For now, don’t worry about handling input on the client, that will come in the next sections. What changes did you have to make to your code to run the server? How about the dummy client? How does your code for this portion of the assignment differ from the code for your remote logger?

In addition, you should implement the Chandy/Misra/Bryant (CMB) algorithm we discussed in class for synchronizing events. This will require you to augment your event system to timestamp all events. What source of time are you using to timestamp your events? What other options are there? How do they compare in terms of performance?

Part 2, Peer-to-peer Multiplayer

Recall that a distinguishing characteristic of peer-to-peer networked architectures is that each participant is both a client and a server. In other words, there are some game objects that a machine is primarily responsible for, and others for which it acts more as a “dummy renderer.” Use your configuration manager to create at least two different, but disjoint, game environments, including a unique character in each (e.g., using different colors). Run each of these environments in a separate version of your code (on a separate machine if available), and exchange objects and updates with each other. Make sure that the union of objects from all peers is visible at all locations.

What changes did you have to make to your code to handle this situation? What does that tell you about peer-to-peer multiplayer games? More importantly, how does event synchronization play into your peer-to-peer game?

---

Part 3, Client-Server Multiplayer

For this part of the homework you are to implement a client-server architecture for your game. In a client-server architecture, the server is primarily responsible for all of the objects in the game world, as well as coordinating updates across the clients. In contrast, each client is responsible for a character object, handling its input, and making local updates that are coordinated through the server. Make sure your client-server setup handles at least two clients, but feel free to attempt more. What scalability issues do you encounter?

What changes did you have to make to your code to handle this situation? What does that tell you about client-server multiplayer games? More importantly, how does event synchronization play into your client-server game?

Extra Credit (20pts Max), Code Integration and Update Protocols

Your task for the extra credit portion of the assignment is twofold. First, you are to create a networked multiplayer implementation of your game using either the peer-to-peer or client/server architecture; however, you must integrate your code so that each client runs different versions of the code and the correct information is sent across the network. In other words, if Sally and Dan work together to implement a client/server architecture, one client will run Sally’s code and the other will run Dan’s code. The server can be either Sally’s or Dan’s code. Some changes may need to be made to both codebases to ensure the proper information is distributed, but all rendering and input management should be handled as closely as possible to previous assignments.

Second, implement a “battle mode” for the multiplayer version of the game. Points are awarded by “bonking” your opponent similar to the Super Mario Brother’s mechanic of killing “Goombas” by jumping on them. Make sure to keep track of points and display them to the screen, and don’t forget to set a win condition!

What design decisions did you have to make to get this working? What was your solution? How did integrating two versions of the code compare to parts 2 and 3 of the assignment where only your code was used?

Submission

In 3–5 pages, please address each of the questions raised in this assignment. Additionally, please discuss how both peer-to-peer and client-server multiplayer games relate to single player games. If you were to set out to design a networked multiplayer engine, how would you design it differently from what you have designed for this class so far?

What to Submit

By 3:50pm on 11/11/11, please upload to moodle a .zip file containing your Processing code and a pdf of your writeup.