Andrew Seba

# Lab Report 07

## Introduction

In this lab we learned about the client and server interactions of variables in a game. With this lab we are able to control client-side prediction and latency compensation.

## Methods

First we did a fully server authoritative game where the server initializes and syncs the states of the players. Every frame the client updates the variables from the server state. The move function for the player sends a key to the server for it to simulate the movement. Using Cmd’s that send the KeyCode the server state will be updated.

Then we did a client-side prediction script where we filled up a queue of keycodes that the client will process using dequeue() for each keycode that supplies movements for other players positions on their game (dummy gameObjects) between server updates.

## Conclusion

Using a queue of keycodes were able to predict the players movement between server updates and fill in for lag.

### Think About It

1. What is the difference between [Client], [Server], and [Command]?

[Client] is a header for a function that only the Client will run  
 [Server] is a header for a function that only the server can call  
 [Command] is a header for a function that will send data to the server.

2. What is a hook and what is it used for?

hook was to have the game update the gameObject variables from the [syncvar] because it doesn’t render the variable changes.

3. What is client-side prediction?

Where the clients will move the dummies with commands from the last update to predict what they are going to do.

4. What is latency compensation?

Making actions work in the client-side prediction model.

5. Why is client-side prediction and latency compensation used?

To make a smooth user experience and playable network games.