

**INFR 3110**  
**Game Engine Design**  
**& Implementation**

**Assignment 1 Report**

Name: Dimitrios Stefanakos  
Student #: 100621470

In this assignment, I have created a DLL plugin for Unity that has the capability of saving an object's position during gameplay, and then loading that position with the press of a button. The player takes control of a cube, who can move it around the scene using keyboard controls, and then save its position. Once saved, the DLL I created gets the position information from Unity and creates a text file with the saved x,y,z coordinates of the player's object position. The DLL file reads the text file and then sends that data back to Unity when the player loads back the previous position of the object.

In my DLL source code, I created a position save function which writes the position data of the object's x, y and z coordinates. When it does, it creates a text file and stores the data there.

```
int SimpleClass::posSave(float cubeXposition, float cubeYposition, float cubeZposition)
{
    ofstream Write;
    Write.open("Position Writer");
    Write << cubeXposition << endl;
    Write << cubeYposition << endl;
    Write << cubeZposition << endl;

    Write.close();
    return 1;
}
```

I also created a Loading function which reads the data from the text file created earlier. The function will read the 3 stored values of the file, and load them back into Unity when the player presses the Load button

### Movement & Saving/Loading Controls

- **W** - Move Forward
- **A** - Move Left
- **S** - Move Backwards
- **D** - Move Right
- **U** - Save button
- **L** - Load button

```
void SimpleClass::posLoad()
{
    ifstream Read;
    Read.open("Position Writer");
    string line;
    holdPos = new float[3];

    holdPos[0] = 0.1f;
    holdPos[1] = 0.1f;
    holdPos[2] = 0.1f;

    if(Read.is_open())
    {
        Read >> holdPos[0];
        Read >> holdPos[1];
        Read >> holdPos[2];
    }
    Read.close();
}
```