

To start, load the unity file and press the “Play” button to test the scene. You can move the cube using “WASD” or the arrow keys.

To save your position, press the “Q” button on your keyboard, and to load your previously saved position, press the “E” button on your keyboard.

```
if (fileStream.is_open()) //if the file opens successfully
{
    fileStream << x << "," << y << "," << z; //saves the x, y and z coordinates to file
    fileStream.close();
    return true;
}
```

This code snippet from the DLL saves the X Y and Z coordinates to a file

```
if (fileStream.is_open()) //if file opens successfully
{
    fileStream >> s;
    fileStream.close();
    if (sscanf(s.c_str(), "%f,%f,%f", &x, &y, &z) != 3) {
        return false;
    } //reads the x y and z coordinates
    return true;
}
```

This code snippet from the DLL reads and uses the previously saved coordinates.

```
2 references
public class SaveLoadWrapper
{
    [DllImport("Assignment_1.dll")]
    1 reference
    public static extern bool saveToFile(string filename, float x, float y, float z);

    [DllImport("Assignment_1.dll")]
    1 reference
    public static extern bool loadTheFile(string filename, ref float x, ref float y, ref float z);
}
```

This code snippet from the save load script tells unity that the DLL has functions that exist, and it can use ‘em.

Then, unity is told to use the functions, and apply that to “boi”, which is the name I gave the variable to represent the player/cube. With that, you just tell it “When I click ‘Q’ I want you to save, and when I click ‘E’ I want you to load”

That’s it. Its as easy as that.

All code was written from scratch, with occasional talking-only aid from Dylan Moore.