**Project 1: Introduction to Working in Real-time**

At the end of this project you will be able to setup a collaborative and visually attractive scene in Unity complete with camera animation. This is a multi-week project divided into four assignments with one part due each week.

Project breakdown and due dates:

* Part 1: Due Thursday, 10/04 – Initial Unity scene
* Part 2: Due Thursday, 10/11 – Camera animation
* Part 3: Due Thursday, 10/18 – Git project setup
* Part 3: Due Thursday, 10/25 – Lighting and post-processing

**TURN IN INSTRUCTIONS HERE:**

Please follow the link and submit your assignments to the correct assignment listing.

[**https://canvas.uw.edu/courses/1260979/assignments**](https://canvas.uw.edu/courses/1260979/assignments)

**Part 1: Becoming familiar with using a game engine**

One of the fundamental skills in using a game engine is building out a scene – this includes placing and transforming geometry, lighting, and creating materials.

Using (at a minimum) the provided assets, build out a basic scene in Unity.

Resources:

**LINK TO CONTENT HERE:** [**https://courses.cs.washington.edu/courses/cse490j/18au/supplementarydocs/ClassOneResources.zip**](https://courses.cs.washington.edu/courses/cse490j/18au/supplementarydocs/ClassOneResources.zip)

This scene must include the follow:

* Four unique meshes
* Two additional light sources *other than* the directional light
* Custom materials assigned to each mesh

You may use any rendering pipeline.

You are encouraged to go beyond the required items and include your own geometry and texture files. Embellish your scene as much as you want; you will have to work in these scenes for the next few weeks so make them fun! <https://poly.google.com> and <https://www.remix3d.com/> are great resources for free models. Please record the models collected at poly.google.com and only use context that is remixable. Remixable content published under a CC-BY license is free to use as long as you credit the author.

Most importantly this scene must look visually pleasing.

Your scene should meet the following requirements:

* Assets are scaled appropriately
* Assets are placed sensibly
* Colors in the scene are balanced
* Lighting suggests a particular time of day
* Lighting and scene colors complement one another
* Shadows are present
* Scene contains enough objects to be visually interesting

What to turn in:

* Three screenshots that showcase the best angles of your scene
* Your Unity project

**Part 2: Adding camera animation**

Using Cinemachine in Unity, create a fly-through of your scene.

Your fly-through cinematic must meet the following criteria:

* At least 15 seconds long
* Smooth camera motion
* No geometry clipping

This is also an opportunity to clean up your projects and make sure they are organized (this will prepare you for part 3). A clean project should have:

* Assets sorted in appropriate folders
* Level organized appropriately
* Assets named logically
* Organized hierarchy
* The level following guidelines covered in class

What to turn in:

* Your Unity project with instructions on how to watch your fly-through cinematic.

**Part 3: Collaboration**

Rarely is a film or game made alone, but a product of collaboration. Revision control (such as Git) is used to make collaboration easier and manageable for multiple users.

Create a **github/gitlab** account and put your project on **github/gitlab**. Your git project must include:

* .gitignore
* Changelog.md
* readme.md

The ReadMe should include instructions for how other users can access your project and what organizational standards/guidelines should be followed when editing your project. It should also include instructions on how your fly-through is set up and can be played. Make sure the ReadMe also credits the authors for any poly.google content that exists in your project.

What to turn in:

* Link to your git project

**Part 4: Lighting and Post-processing**

Lighting combined with post-processing is important for conveying visual tone in a scene. For this assignment you will edit another student’s project by animating the lighting and post-processing in order to change the tone of the scene during the cinematic fly-through.

Steps:

1. Fix any existing issues in the current lighting setup of the project you were assigned.
2. Setup post-processing in the project.
3. Edit the fly-through cinematic to animate the lighting in a way that transitions to a completely different time of day from the beginning to end of the sequence.
4. Animate at least one post-processing layer during the cinematic as well. You may animate as many post-processing layers and properties as needed to create the tonal effect you are going for.
5. Update the git project with these changes.
6. Use helpful commit messages.
7. Update the changelog with info about what you changed, and update the ReadMe if necessary.

What to turn in:

* Link to the git project you have updated

**Grading Rubric Part 1 (100 points total):**

* 20 – mesh requirements met
* 20 – material requirements met
* 10 – light requirements met
* 40 – scene appearance (how visually attractive your scene looks)
* 10 – screenshot quality (how attractive are the screenshots you submitted)

Note: all grading requires working Unity projects at time of submission. **Make sure you submit a working project**. Include a ReadMe.txt if there are specific requirements for using/opening your project.

**Grading Rubric Part 2 (100 points total):**

* 20 – cinematic/fly-through is easy to run
* 60 – quality of cinematic/fly-through
* 20 – quality of project organization

**Grading Rubric Part 3 (100 points total):**

* 20 – .gitignore is correct
* 10 – changelog is present and thorough
* 30 – ReadMe is thorough and helpful
* 40 – following instructions in the ReadMe results in a working project

**Grading Rubric Part 4 (100 points total):**

* 5 – changelog is updated
* 5 – commit messages are helpful
* 10 – git project still works when following ReadMe instructions
* 20 – cinematic/fly-through showcases an obvious tonal change
* 20 – lighting is setup correctly and shadows are crisp and believable
* 20 – post-processing is setup correctly
* 20 – quality of cinematic after lighting and post-processing animation