Games Engines 1 Assignment 2021-2022

Please include a detailed readme.md in your git repo. Here is a template. In order to help me grade the assignments include in your readme:

- A detailed description of what your assignment does and how it works
- Which parts of the assignmemnt you developed yourself vs parts that come from the examples we made on the course or that come from tutorials
- What you are most proud of about the assignment
- Instructions for building and running (if necessary)
- Include an embedded youtube video of the assignment in the readme.md and also submit the Youtube video link in the submission form. If you are on Windows 10 you can press Windows Key, Alt and R to take a video. This is how you can capture videos if you are on a Mac. Also you can use OBS.

This is an example of how you can embed a youtube video in a markdown file:

```
[![YouTube](http://img.youtube.com/vi/VIDEO_ID/0.jpg)](https://www.youtube.com/watch?v=VIDEO
```

For example, this URL:

https://www.youtube.com/watch?v=ii049d7UFrg

Becomes:

```
[![YouTube](http://img.youtube.com/vi/ii049d7UFrg/0.jpg)](https://www.youtube.com/watch?v=ii
```

Due dates:

- Week 5 - Proposal & git repo
- Week 11 - Final submission & in-class demos

Assignment brief

Your assignment is to create an impressive, beautiful and awe inspiring system in Unity by using procedural generation. Paint a picture in code. Your aim is to wow me. It should look like a tech demo or some of the beautiful visuals that you might see at an electronic music event. Check these out for examples:

2020-2021 Assignment playlist

2019-2020 Assignment playlist

You should aim to create the effect of your project through produral means and I encourage you to make everything in this assignment yourself. If you want to use 3D models you should create them yourself. The only exception is audio. If you find a beautiful piece of music that you need for your demo you can go ahead and use it.

Also check out the Procedural generation reddit forum

Here are some ideas but please don't limit yourself to these! I would prefer you to come up with your own:

- A Perlin noise flow field "flocking" simulation
- Marching cubes cave simulation
- A Procedural Minecraft style voxel world, with Mountains and rivers and lakes, coloured and rendered appropriately
- Use the Physics engine in Unity to make a Procedural creature, Robot or Machine
- A Procedural planet
- Use the Mechanim system in Unity to rig and animate an animal or robot character and control the animations with a state machine
- A geometric structure based on Fractals, noise or mathematical functions such as sine waves etc
- An infinite terrain or building
- A procedural city generator
- Procedural trees or forests
- Something Kinect or Leap motion controlled. Or use a custom controller that you make yourself

- A simulation of a natural or biological system
- A network simulation
- Animate some system using mathematical functions
- Something controlled by music or sound

You should use good OO practices aligned with the GameObject/Component model that Unity uses. In other words, use inheritance, polymorphism, abstract classes and interfaces to implement game components. It should be easy for someone to customise your creation and you should expose sliders and controls and display appropriate gizmos.

Marking Scheme

Category	%
Complexity	30%
Polish	30%
Code design	30%
Jazz	10%

Rubric

Grade	Profile
First	You will have spent in excess of 20 hours on the assignment. There will be 80+ git check-ins. Your demo will look impressive and or beautiful and or very novel or cool. The demo should be technically impressive and you will have used some complex maths, algorithms, challenging API's or demonstrate some deep knowledge of Unity in your solution. Your system will be exposed to Unity properly via game components, gizmos and editors. It should be easy for someone to customise. It will be of the quality of a €10 asset from the Unity Asset Store
2.1	You will have spend 10-16 hours on the assignment with around 30-70 check-ins. You will have implemented a moderately complex system which relies somewhat on Unity stuff like the animation system, trail renderers or particle systems to be impressive. In other words its got a bit more emphasis on stuff thats built into Unity rather than what you made. Nevertheless you should have implemented some complex system. You might have made some game components and classes, but your solution might have always used best practices and design patterns. For example, long methods here and there without enough granularity. You might not have exposed all the editors and gizmos that your system might benefit from and your system might be not so customisable
2.2	You will spend around a day on the assignment and will have implemented a simple system based on sine waves or similar. You might have around 30 check-ins or less. Your solutions is one or two game components in complexity. There might be random colours no audio or inappropriate audio. Your system might glitch a bit and would need a fair bit of future work to make it publishable or useful

Grade	Profile
Pass	You will spend less than a day on the assignment and get something very basic implemented like a spiral or a simple voxel world. No git use or evidence of many check-ins
Fail	Nothing works, and no git. Looks like it took a few hours to make