Project 2 Urban: Buildings
CS 4803 PG, Spring 2018

In this project you will create buildings using a procedural approach. You will select a category of buildings, and your program will be able to make a variety of buildings from this category. Some possible category of buildings include apartments, row houses, log cabins, Swiss chalets, houses, or skyscrapers. As with all of the projects in this class, you will create the geometry of your buildings using triangle meshes.

Due Date and Grading

This project is due on Friday, March 9 at 11:55pm. Each day late after Friday will cause your grade on the project to drop by 5%, and we will not accept the project more than three days beyond the due date. This project will have a 10% "effort" component. If your project meets all of the listed criteria, you will get at least 90% credit on this warm-up project. The final 10% of your grade will be based on our judgment of whether you showed creativity and care in your project.

Authorship Rules

Each student must work on this assignment on their own or with one other student who is in the class. Seek out the TA's and the instructor for help with Unity. You may **not** use code from other sources, including code from the web, from videos, from Github, from books, or from other people than your one class partner for this project.

Project Description

This project is designed to be entirely separate from the earlier projects in this course. You do not need to combine your buildings with your terrain or street projects.

Here is a list of the required elements for the project:

- Automatically create a variety of buildings.
- When you your program is run, show at least three buildings side-by-side, with the differences between them due to the use of random numbers.
- Have an entry box for the initial random seed for your buildings, so that a user can cause your program to create different buildings.
- Your buildings must exhibit a variety of **footprints** (chosen from at least six different footprints). The footprint variation must not simply be longer or shorter rectangles. Some of the footprints must be concave.
- Your buildings must show variations in **elevation profiles**, and this must include the possibility of one part of a building being taller than another part.
- There must be at least two different styles of doors and two different styles of windows for the facades of your buildings. See the notes below on façade geometry.
- Your buildings must demonstrate at least two different **roof styles**. These roof styles should join properly where different building segments cross (e.g. cross-hip or cross-gable).
- The walls of your buildings must come in at least two **colors or patterns**.
- At least one visual element of your buildings should be created through the use of **texture maps**.

You may choose to implement your procedural buildings in any manner that you wish. Our recommendation is that you use a 2D grid of cells to specify the footprint of a building, and use per-cell integers to describe the elevation profile. Although many of the published building construction papers use formal grammars to construct buildings, we recommend that you do **not** take this approach. Grammar approaches to creating buildings are quite difficult to implement, and if you use a grammar you are likely to find you cannot finish all of the project requirements in time.

The creation of your buildings footprints does not need to be a complex algorithm. You may simply have a small "library" of different footprint shapes that are pre-specified, and randomly choose from among them.

Your façade elements (windows, doors, etc.) should be in exactly the same plane as the walls that they are a part of. You must **not** place these elements slightly out from the wall geometry. This means that you will need to build each unit of the wall in rectangular pieces so that they properly surround the door and/or window elements of the façade.

Additional Requirements for Two Person Teams

If you wish, you may team up with one other person on this project, but this must be someone you have not yet worked with for this class. If you are working in a two-person team, your project must meet additional requirements beyond those for a one-person team. Here are the additional requirements:

- Your windows and doors must not be just flat polygons, but must include 3D geometry such as a door knob, steps, a frame or a window sill.
- Your buildings must incorporate at least two different 3D geometric structures that are distinct from typical façade elements. Possible examples are: rooftop water towers, air conditioning units, fire escapes, chimneys.

Possible Additions (Not Required)

Below are some ideas about possible additions to your buildings. These are not required elements, but are ideas that you may consider if you feel inspired to go beyond the project basics.

- Create overhangs such as carports or a roofed porch that is supported with pillars.
- Incorporate curved elements such as cylinders into your building styles.

Additional Rules

As with each of the projects in this course, all of the objects in your scene should be created by you from within Unity. You should not include game assets that have been made using other programs (Maya, Blender, etc.), nor should you include assets from the Assets Store. The one exception to this rule is that you can load texture image assets (including a sky box). You may wish to create such textures using a paint program or by taking photos.

Turning in the Project

You will turn in your project on T-square. To do this, first create an empty folder that is called your name. Next, determine where Unity stores your projects, find the directory that contains your warm-up project, and copy its two sub-directories called Assets and ProjectSettings into the empty folder. Zip up the folder that contains Assets and ProjectSettings, and submit this zipped file to T-square. Do not create a .rar or .tar file – please create a .zip file. If your zipped file is more than a few megabytes in size, you probably copied too many sub-directories, and in particular the Library sub-directory should **not** be included because of its large size.