

Project 1 Urban: Street Creation

CS 4803 PG, Spring 2018

In this project you will write code that will automatically create a large pattern of streets. You can either use a grid-based method of street layout (recommended), or use a more free-form layout algorithm (much more challenging). Your street components will be made of triangle meshes, and you will add visual detail using materials and/or textures. The notion of “street” should be interpreted broadly as being any kind of pathway for vehicles, humans or animals to travel along.

Due Date and Grading

This project is due on Friday, Feb 16 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 street network 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 should feel free to use the provided C# code as a starting point for creating your project. 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

Your main task is to create a network of streets. The term “street” should be interpreted broadly for this assignment, and any variations such as these are fine: dirt roads, walking paths, train tracks, subway rails, raised boardwalks, suspension bridges. If you are unsure whether your idea for a “street” is okay, check with the instructor.

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

- Automatically create a pattern of streets.
- Your street pattern must contain at least one example of each of the following street components: straight pieces, turns, 4-way intersections, T-junctions, and dead ends.
- Your street components must be made from non-trivial polygon meshes. At least one of your street components (bend in road, T-junction) should be a mesh that contains at least 32 vertices.
- Add **color variations** to your roads using differently colored meshes, textures, or both.
- Place at least six **simple buildings** in your street map (one or two primitives per building).
- Allow camera movement, as in the provided sample code.

Each of the streets in your pattern should gracefully and neatly join with other streets. That is, intersections should be specially created to properly allow the geometry of the two streets to come together without breaks. Just drawing one street on top of another is not a proper intersection.

Because streets are primarily flat, it is fine if the meshes for your street components are not “solid” objects. You may even decide to layer several flat meshes on top of each other. For example, you might use rectangular meshes to help add a broken yellow line to divide your road in the middle. If you do decide to put flat elements on top of each other, you will have to experiment with how far you raise the top mesh so that it reliably is drawn above the lower mesh.

If your “streets” are not city streets but are elements like railroad tracks or suspended subway rails, your geometry might indeed be long, solid objects. This is absolutely fine. Your streets do not need to be flat on the ground in these cases.

In order to make sure your code shows off all of the required elements (e.g. T-junction, dead end), you may want to make a public variable that is a random number seed that is used to initialize the random number generator. Then you can search for a random number seed that properly shows off all of the required elements.

You don’t have to place too many buildings into your scene. Six buildings is fine to get full credit for this required component of the project. Of course, your buildings should not be placed on top of any streets. Your buildings must be simple, placeholder buildings that are made of just one or two Unity primitives, without texture. You can create much more elaborate buildings in a later project. Please do not use a custom mesh for any of your buildings.

It is relatively easy to create terrain on demand when the camera is moved near the edge of the terrain. It is much more difficult, however, to create additional street components on demand that nicely meet up with existing streets. For this reason, you should create a single **large** pattern of streets once, when your code is first run. Creating new streets on demand is **not** a part of the required elements for this project.

Additional Requirements for Two Person Teams

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 street pattern must include highways that are at least twice as wide as your regular roads. These highways should smoothly join with each other and with the regular roads.
- Your street network must include bridges that span water features such as rivers.

If you are using a grid-based system for making your street pattern, your highways must be at least two grid cells wide.

Your bridges must create geometry that rises above the street level. That is, your bridges cannot just be flat geometry that are drawn on top of the water features – they must rise up in the vertical direction.

Possible Additions (Not Required)

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

- Place additional geometric elements in your scene, such as fire hydrants or street signs.
- Create street elements such as a roundabout or a cloverleaf.
- Add parks, rivers or ponds that your streets avoid.
- Create more streets on demand (very difficult to do well).

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. You may wish to create such textures using a paint program or by taking photos. You can use textures to add variety to your street components, but you should not place textures on your simple buildings.

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 megabyte in size, you copied too many sub-directories, and in particular the Library sub-directory should **not** be included because of its large size.