

Project 3 Wilderness: Creatures

CS 4803 PG, Spring 2018

For this project you will write code that will automatically create a variety of creatures. You will decide on a category of creature, real or fictional, and your program will make a variety of creature bodies from this category. Some possible categories of creatures include: birds, dragons, insects, crustaceans, sea monsters, mammals, fish, mollusks, and dinosaurs. When given a new random seed, your program will create different creature bodies from the given category. A central part of this assignment is for you to program (and use) a technique for curved surface creation. Possible choices for this include: Bezier or Catmull-Rom tubes, Bezier surfaces of revolution, subdivision surfaces, cubic Bezier patches, and implicit surfaces.

Due Date and Grading

This project is due on Friday, April 6 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 18% “effort” component. If your project meets all of the listed criteria, you will get at least 82% credit on this project. The final 18% 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 creatures with either of your first two projects.

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

- **Automatically** create a variety of creatures. The user should not have to make any choices.
- When you your program is run, show at least five creatures side-by-side, with the differences between them due to the use of random numbers.
- Have an entry box for the “master” random seed for your creatures, so that a user can cause your program to create different creatures.
- Your creatures must exhibit variety in terms of **swappable parts**. At least two body parts for your creature must have swappable parts, and you should have at least three variations for each.
- Your creatures must incorporate some form of **curved surface**, for which you will write the code.
- Your creatures must show variety in terms of the **sizes** of various body parts.
- Your creatures must show some variety in terms of their **colors**. The color variation may be for the overall color of the creature, or variations in color between different body parts.

It is important that your program create the creature bodies entirely automatically. Your user should not have to do anything beyond entering a single random number seed in order to get several new creature bodies. The random number seed that the user enters should be a “master seed” that is used to create a short list (e.g. five) of per-creature random number seeds. Then the creatures that you show should be created based on these per-creature random number seeds.

A key requirement of this project is that you write the code to create curved surfaces. Simple curved surfaces such as cylinders, cones, sphere and ellipsoids do not count for this. Your choices of curved surfaces include: Bezier and Catmull-Rom tubes, Bezier surfaces of revolution, subdivision surfaces, cubic

Bezier patches, and implicit surfaces. If you are using Bezier curves to make tubes or surfaces of revolution, you must use quadratic or cubic curves (not linear curves). If you have a different idea for curved surface creation, check with the instructor about it. You must write the code for one of these types of curved surfaces yourself. You must also use this curved surface type in creating your creatures. You may use your curved surface to help make up the main body of the creature (e.g. the torso), you may use it to create some of the swappable body parts of your creature, or you may use it in both of these ways.

Another important aspect of this project is to have different variations of body parts, and to be able to mix-and-match these parts to form a variety of creatures. For example, you might create several variations of predator claws, and have different of your creatures use different kinds of claws. Some possible parts that you could create variations for include: wings, arms, legs, tails, fins, claws, feet, hoofs, head, dorsal crest, ears, eyes, nose, mouth, antenna, tentacles, gills, and beak. At a minimum, you must demonstrate swappable parts for two different creature body parts (e.g. eyes and ears), and show at least three variations for each of these. Remember that the “effort” component is a large part of your grade, and the more variety that you show, the better.

The swappable body parts that you create do not need to be a single mesh. Similar to the first warm-up exercise, you may want to piece together different meshes using transformations to create a more complex GameObject. For example, you might use a simple polygon mesh for the shape of a paw, and you might add claws to this using cones or Bezier tubes. An eye might consist of several spheres of different colors, such as white for the sclera, with a small portion of a blue sphere poking out to make up the iris. You can assemble these different parts either by hand (creating and dragging GameObjects into a hierarchy), or you can do this using code. Either option is fine. The individual pieces of a given body part can be a triangle mesh, a primitive, or a curved object that you created using your own code.

Some of the parts for your creatures must show off variation in sizes. For example, a specific leg type for a lizard might take on a variety of different lengths. This is **not** the same as the variation between swappable parts. That is, you should not create three different pre-made legs of different sizes. These variations in size should be continuous, that is, guided by floating point numbers. These size differences can be made through programming when you create the mesh, or by careful use of scale transformations. Take care to keep your body part sizes within reasonable ranges for the types of creatures that you are making.

It is very difficult to make smooth joins between different body parts of a creature. For example, it can be difficult to smoothly attach an arm to a torso. You do **not** have to make smooth joins between different body parts. You are allowed to simply overlap the two body parts enough so that the end of one (such as an arm) is buried in another (such as the torso).

There are a few types of creatures that are not complex enough to be good choices for this project. In particular, you are not allowed to use snakes or worms as your creatures because they simply have too little variations in their body parts. If you are uncertain about whether a given category of creature is suitable for this project, seek the advice of the instructor.

Effort

Part of the grade for this project is our judgment of “effort”. If you take short cuts in order to just barely address each of the required elements, your effort score will be low. There are several factors that we will take into account in judging effort: variety, detail, cleanliness, and creativity. *Variety* refers to the visual differences between the creatures that you create. If they all look like slight variations of each other, this does not show much variety. More variety in creature body parts will help here. *Detail* refers to whether you took care to include visual elements that add richness to your creatures. *Cleanliness* is whether your creature’s body parts are pieced together cleanly or sloppily. For example, are the open ends of a tube clearly visible (which is undesirable), or are they neatly hidden in the body part from which the tube extends?

Note that most creatures have rounded body parts. If you simply use a collection of boxes and cylinders to build your creature bodies, the results will not look nice because the geometry will have unrealistically sharp edged. This is an issue of the detail of the creature body parts.

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:

- You should implement a second method of creating curved surfaces, and use this method to create some of the parts.
- You should demonstrate a greater variety of swappable parts than a single-person project. In particular, at least four body parts for your creature must have swappable parts, and you should have at least three variations for each.
- The main body part for your creature (e.g. the torso) should be one of its swappable parts.

Possible Additions (Not Required)

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

- Add textures to your creatures.
- Cause your creature to move around a little while it is standing or sitting in one place.
- Add geometric surface details such as scales, fur or feathers to your creature. (This can be very difficult!)

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. During this project, you may be tempted to use another program to create polygon meshes for your creature. Do not do this! 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 also create such textures using a paint program or by taking photos.

You may wonder how you can determine the coordinates for different parts, if you cannot use an external modeling program. You have several choices. You can get out graph paper, draw out your parts, and copy the positions into your program based on the grid positions. You may use a ruler or a yard stick and measure real objects to get distances and locations. If you are ambitious, it is entirely okay to make a mini-editor inside of Unity, so long as the code is entirely your own. For instance, you may wish to create a Bezier or Catmull-Rom curve editor that allows you to interactively place control points. Unity provides tools called Gizmos and Handles that can help you create such a mini-editor. Be warned, however, that this will take a lot of work. Using graph paper is much easier.

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 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.