MP1: Getting to know the tools and weekend entertainment

New Attempt

Due Oct 9 by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip
Available after Sep 29 at 11:59pm

Objective

In this programming assignment, we familiarize with the learning tool, Unity, that we will be using for the rest of this quarter. You should also use this opportunity to:

- Evaluate how comfortable you are with the kinds of learning that is expected [mainly on how to use Unity as a tool].
- Ensure that you understand exactly what is involved in submitting a programming assignment.

This assignment should take you no more than 5 or 6 hours to complete. Most of the time should be spent on asking and looking for answers to: "*how do I make Unity do …*"

Approach

Please refer to **my attempt at the solution** (https://canvas.uw.edu/courses/1492345/files/81191474/download?download_frd=1) (download, unzip to run). Initially, your world consists of a **CreationPlane** and a **CreationTarget**:

- **CreationPlane**: is a **Quad** with
  - Transform:
    - Position: 0, 0, 0
    - Rotation: 90, 0, 0 (such that the Quad is on the x/z plane)
    - Scale: 14, 14, 14
  - Note: the corners of the plane spans between –7 to +7
  - If you choose to work with a **Plane**, your transform settings will be different, but, it is important to make sure the size and location of the CreationPlane to be specified as above.
- **CreationTarget**: is a sphere with
  - Transform:
    - Position: user defined!
    - Rotation: 0, 0, 0
    - Scale: 0.5, 0.5, 0.5

- Color: black

You must also support a drop down menu that provides the options of creating a Sphere, Cube, or Cylinder. Here are the specification for each of the shape:

- Sphere:
  - Size of 1 (scale = 1)
  - Speed = 1 unit **per second**
  - No rotation
  - Moving direction x-axis
  - Range: 0 < x < 5
  - Color:
    - When traveling in the positive-X: color= (1, 1, 1)
    - When traveling in the negative-X: color=(0, 1, 1)
  - Cube:
    - Size of 1 (scale = 1)
    - Speed = 1 unit **per second**
    - Rotation speed = about y-axis, 90-degrees per second
    - Moving direction y-axis
    - Range: 0 < y < 5
    - Color:
      - When traveling in the positive-Y: color=(1, 1, 1)
      - When traveling in the negative-Y: color=(1, 0, 1)
    - Cylinder:
      - Scale = (1, 2, 1)
      - Speed = 1 unit **per second**
      - No rotation
      - Moving direction z-axis
      - Range: 0 < z < 5
      - Color:
        - When traveling in the positive-Z: color=(1, 1, 1)
        - When traveling in the negative-Z: color=(1, 1, 0)

Behaviors

There are three basic behaviors that you must support.

1. **Shape creation:** When the user selects a shape-item from the drop down menu, you should create a new corresponding shape at the location defined by the **CreationTarget**. In all cases, a newly created shape should be resting on the **CreationPlane** and attempt to travel towards the position direction. If the creation position is larger than the valid range of 5, the created shape will then travel in the negative direction. After a shape's first change of traveling direction, it should never cross the 0 to 5 range.

2. **Creation target manipulation:** You must support Left-Mouse-Button click (LMB):

- When LMB is over the **CreationPlane** and **not** over any other objects, the **CreationTarget** is moved to the mouse click position. **Note:**
  - CreattionTarget must always be entirely above the CreationPlane.
- CreattionTarget must always be entirely above the CreationPlane.

3. **Object deletion:** LMB over a created object should delete the object. **Note:** one can never delete the **CreationPlane** or the **CreationTarget**.

Hints:

1. For dropdown menu, I googled "*unity API dropdown*". Here is the API documentation I read when learning about drop down menus: **https://docs.unity3d.com/ScriptReference/UIElements.DropdownMenu.html (https://docs.unity3d.com/ScriptReference/UIElements.DropdownMenu.html)**
   - I worked ***OptionData*** and ***OptionDataList*** classes and initialize the menu items in my code in my first try and then realize the item initialization can be done through the editor.
2. I googled "*unity elapsed time*" to try to find out how to work with ***per-second*** movements and rotations. Take a look at ***deltaTime*** (**https://docs.unity3d.com/ScriptReference/Time-deltaTime.html (https://docs.unity3d.com/ScriptReference/Time-deltaTime.html)**),
3. To avoid mouse-click selecting the **CreationTarget**, I place the corresponding sphere into a different ***Layer***, and *Raycast(ray, hitInfo, **layer-1**)* [the default layer] to click select floor or the created objects.
   - If mouse click is on the floor, I move the CreationTarget over to the (x, someValue, z) where x, and z are from the mouse-click.
   - Do a google on "*unity layer*" to find out more about Layers.
4. Take a look at *Rotate(Axis, degree)* to rotate a GameObject

Credit Distribution

1. **(5%)** Initial frame: **CreationPlane** and **CreationTarget**
   - (5%) Correct size/color for both
   - (5%) Both entirely visible from the runtime camera
   - (**-50%**) **If the CreatePlane is not entire visible at 1290x768**

2. **(25%) CreationTarget** interaction
   - (10%) Proper placing of **CreationTarget** at LMB (place anywere on the CreationPlane)
   - (5%) The sphere drawn always above **CreationPlane**
   - (10%) Can never delete **CreationTarget**

3. **(20%)** Dropdown menu support
    - (5%) Menu title: "Object To Create" at all time
    - (10%) Shows the three primitive to create
    - (10%) Able to create and display the primitive

4. **(20%)** Created Primitive Behavior
    - (5%) Create at the CreateTarget position with entire primitive above the CreationPlane
    - (15%) Movements: Sphere:X, Cube:Y+Rotate, Cylinder:Z

5. **(15%)** Click to delete:
    - (10%) Proper deletion of mouse clicked primitive
    - (10%) Do not delete the CreationPlane and CreationTarget

6. **(15%)** Submission and Others
    - (10%) Zipped all source code + exe folder
    - (10%) Proper zipfile naming: FirstLastNameMP1.zip

Submission:

- **Source code**: just the three folders: **Assets**, **Packages, ProjectSettings**
- **Executable**: Include a separate **EXE** folder, and build your executable into this folder.

This programming assignment counts *4%* towards your final grade for this class.

Extra Credit:

You can do 100 million (and more) different things to test/learn/demonstrate your knowledge of Unity. I will give out up to 10-points of extra credits on any experiment/features you implement (must be working)!

Extra credit ideas: it is **NOT** as interesting to spend time making your mp "***beautiful***" or do what we can do but just more, e.g.,

- Less interesting to allow the creation of more primitive types (more of the same), we already know how to do this
    - Instead: can you create a car from the drop down menu? [something different]
- Less interesting to place a nice looking geometry for **CreationTarget** (again, we know how to do this)
    - Instead: support multiple **CreationPlane** in different dimensions (make sure I know how to test the basic functionality)
- Other interesting things:
    - Global behavior control: click to stop all movements

    - Individual object control: stop/start individual objects [e.g., with an extra deletion button], or customizable movements
- You can experiment with Unity's Scene support and create separate scenes for your different test cases and extra stuff

As I said, there are 100 million different things you can do.

WATCH OUT: This is due on next week. My plan is to grade your submission and return them to you on following week so that you know what is expected of you!! MP2 is coming on next week. Start early, DO NOT fall behind, it will be difficult to catch up.