CS460 Fall 2022
Name: Shen xie
Github Username: shenxie905
Due Date: 09/26/2022

Assignment 3: Three.js Cubes ... and other geometries

We will use Three.js to create multiple different geometries in an interactive fashion.

In class, we learned how to create a `THREE.Mesh` by combining the `THREE.BoxGeometry` and the `THREE.MeshStandardMaterial`. We also learned how to *unproject* a mouse click from 2D (viewport / screen space) to a 3D position. This way, we were able use the `window.onclick` callback to move a cube to a new position in the 3D scene. Now, we will extend our code.

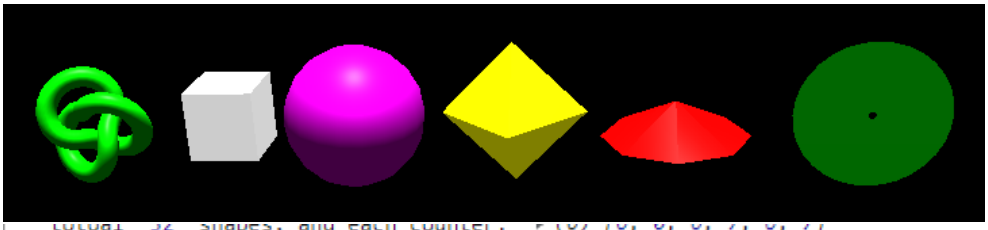
The goal of this assignment is to create multiple different geometries by clicking in the viewport. This means, rather than moving an existing mesh, we will create new ones in the `window.onclick` or `renderer.domelement.onclick` callback. On each click, our code will randomly choose a different geometry and a random color to place the object at the current mouse position.

We will be using six different geometries. Before we start coding, we want to understand their parameters. Please complete the table below. You can find this information in the Three.js documentation at `https://threejs.org/docs/` (scroll down to Geometries). In most cases, we only care about the first few parameters (**please replace the Xs**).

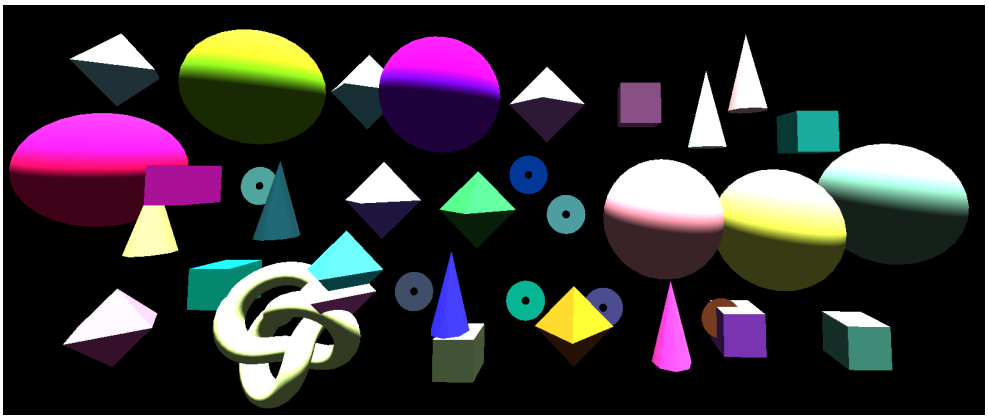
Constructor	Parameters
THREE.BoxGeometry	(width, height, depth)
THREE.TorusKnotGeometry	(X, X, X, X)
THREE.SphereGeometry	(X, X, X)
THREE.OctahedronGeometry	(X)
THREE.ConeGeometry	(X, X)
THREE.RingGeometry	(X, X, X)

Please write code to create one of these six geometries with a random color on each click at the current mouse position. We will use the `SHIFT`-key to distinguish between geometry placement and regular camera movement. Copy the starter code from `https://cs460.org/shortcuts/08/` and save it as **03/index.html** in your github fork. This code includes the `renderer.domElement.onclick` callback, the `SHIFT`-key condition, and the `unproject` functionality.

After six clicks, if you are lucky and you don't have duplicate shapes, this could be your result:



```
totoal  33  shapes, and each counter:  ▶ (6) [6, 0, 6, 8, 6, 7]
totoal  34  shapes, and each counter:  ▶ (6) [6, 0, 6, 9, 6, 7]
totoal  35  shapes, and each counter:  ▶ (6) [6, 1, 6, 9, 6, 7]
totoal  36  shapes, and each counter:  ▶ (6) [7, 1, 6, 9, 6, 7]
```



I got all the six different shapes after 35 clicks.
Please make sure that your code is accessible through Github Pages. Also, please commit this PDF and your final code to your Github fork, and submit a pull request.

Link to your assignment: `file:///C:/Users/xiesh/cs460student/03/index.html`

Bonus (33 points):

Part 1 (5 points): Do you observe Z-Fighting? If yes, when?

YOUR ANSWER: Yes, Z-Fighting happens when two face overlap.

Part 2 (10 points): Please change `window.onclick` to `window.onmousemove`. Now, holding `SHIFT` and moving the mouse draws a ton of shapes. Submit your changed code as part of your `03/index.html` file and **please replace the screenshot below with your drawing**.

Part 3 (18 points): Please keep track of the number of placed objects and print the count in the JavaScript console. Now, with the change to `renderer.domElement.onmousemove`, after how many objects do you see a slower rendering performance?

YOUR ANSWER: After about 2000 objects I saw a slower rendering performance on my computer.

What happens if the console is not open during drawing?

YOUR ANSWER: When the console is not open during drawing, we can't see the log messages in the console, such as how many objects have be drawn.

Can you estimate the total number of triangles drawn as soon as slow-down occurs?

YOUR ANSWER: Yes. When I drew 2000 objects, there were 309 Box BufferGeometries, 327 TorusKnot BufferGeometries, 340 Sphere BufferGeometries, 352 OctahedronBuffer Geometries, 317 ConeBufferGeometries, and 355 RingBufferGeometries. THREE.BoxGeometry(10,10,10) has 12 triangles. THREE.TorusKnotGeometry (10, 3, 100, 16) has 100*16*2=3200 triangles. THREE.SphereGeometry (15, 32, 16) has 32*(16-1)*2= 960 triangles. THREE.OctahedronGeometry(10) has 8 triangles. THREE.ConeGeometry(5,20) has 16 triangles. THREE.RingGeometry(1,5,32) has 32*2*2=128 triangles. Therefore, the total number of triangles are 309*12 + 327*3200 + 340*960 + 352*8+ 317*16 + 355*128= 1429836.