3. a) Explain the behavior of Time.deltaTime in Unity in a few sentences. Add a C# script that associates Time.deltaTime to your primitive object (The specifics of the implementation of Time.deltaTime are left to you, and you may choose to implement it as you see fit). [4 pts]

- Time.deltaTime basically tells the developer how long each frame is taking to execute. Using this information, a developer is able to multiply Time.deltaTime by another value which makes the game framerate independent. A framerate independent game means that the game will run the same on both slower and faster computers regardless of what the system is capable of. It essentially acts as a balancing mechanic so the user with a faster computer doesn't have an advantage.

b) Watch the video of the racing game, Forza Horizon: https://www.youtube.com/watch?v=WtuBLc3cU-o. Based on your understanding of Time.deltaTime, please try to explain where it may have been used in the game. Also, explain how the user experience would vary depending on the machine if it weren't used. [3 pts]

- Time.deltaTime was likely used in helping the car movement animations. In order for the cars to move similarly across machines with varying frame rates, it would need the cars to be based on time rather than framerate so the cars move the same across different platforms.
- If it wasn't used, it would have likely been a huge experience difference between more efficient machines versus the lower end. For instance, on higher end machines, the cars likely would've felt faster and the physics could act differently. This would ruin the user experience because the game would not feel as realistic. Lower end machines would also feel the effects of this because the frame rate would make the cars feel less responsive and have rigid movement.

5. Explain the following:

a) Mesh Renderer [1 pt]

- The Mesh Renderer is a component that allows the camera to see a 3 dimensional object in the scene. It can be used to render characters, environment objects, and other 3 dimensional assets. If it is turned off, it masks the ability to view 3D assets.

b) Box collider [1 pt]

- A box collider is a Unity component that defines the hit box for a GameObject that allows collision detection and physical interaction between other GameObjects in the scene.

c) Input.GetAxis method [1 pt]

- Input.GetAxis is a method in Unity that gives a value of a virtual axis which allows the game to detect user input such as a controller or keyboards. The two axis names are Horizontal and Vertical to map out the input controls for the game.

d) Rigid body [1 pt]

- The Rigid body is a component that allows a GameObject to utilize Unity’s physics engine where GameObjects can have realistic behaviors such as collisions, gravity, and other forces.