Homework\_Two

Written Responses

1. Create a new Unity project named Homework\_Two. [2 pts]
2. Add the “cube” primitive object to your game and name it “Player”. Create a new material and set its color to red (any shade of red is fine). Assign the material to the primitive object to ensure its color is the same as the color defined in the material. [5 pts]
3. 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]
   1. ***The behavior of Time.deltaTime is very important, as it has the ability to tell how long each of the frames takes to execute. It is representative of how much time has passed in the game since the last frame update, which is useful for making movements or actions “frame rate independent.” When a number is multiplied by Time.deltaTime, that causes the game to become “frame rate independent,” which is useful when wanting the game to run at the same speed on different computers.***
   2. 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]
      1. ***Time.deltaTime may have been used in the Forza Horizon game to ensure that there is consistent speed of the car across different frame rates and to make sure that the movement of the car is working smoothly. As talked about with the behavior of Time.deltaTime, this is important for the game/car’s movement because it allows for fluidity even across different computers that have varying frame rates. If Time.deltaTime was not used, the movement of the car would be entirely dependent on the frame rate, which would make it inconsistent across different computers (due to differences in frame rates because some computers are faster/slower than others). This would make the car’s movement faster on higher framed computers and slower on lower framed computers, which would make the game inconsistent and overall unfair.***
4. Add the Cinemachine package to your game. [2 pts]
   1. Add a simple terrain to your game, such as a plane, and make sure that the primitive object is on the plane (and that no part of the object is underneath the plane). [2 pts]
   2. Create and add a Cinemachine virtual camera to the game. Set the virtual camera to follow the primitive object as it moves on the terrain. [3 pts]
5. Explain the following:
   1. Mesh Renderer [1 pt]
      1. ***The “mesh renderer” is a Unity component that allows for 3D objects to be displayed in the scene. It uses materials assigned to it to show the 3D model of the object and can also apply certain lighting and shades to make it visible inside of the game (rather than a 2D object which cannot be seen).***
   2. Box collider [1 pt]
      1. ***The “box collider” is a type of collider in Unity that defines the physical boundaries of an object (which is based on a specific box shape). It is used to be able to tell if two objects are combining or overlapping in the game and is useful for things such as triggers or physical interactions in the game.***
   3. Input.GetAxis method [1 pt]
      1. ***The “Input.GetAxis” method in Unity is a method that returns the value of a virtual axis, which can be used to be able to detect input from devices such as keyboards, game controllers, and joysticks. The names of the axes (horizontal and vertical) are used to map the input controls to actions inside the game.***
   4. Rigid body [1 pt]
      1. ***The “rigid body” is a component in Unity that allows a GameObject to act under the control of Unity’s Physics engine. This allows it to enable realistic physical behaviors such as gravity, collision, and forces.***