For the purpose of scoring, we will use the previously created touchpad control scene, driving a small cube, when the cube touches the sphere it should increment points by 10.

Please refer to the TouchpadControl tutorial before starting this tutorial

1. Lets make the player cube smaller. To do that change the scale on the player (under the transform component) to 0.5 x 0.5 x 0.5.
2. Create a game object → 3D → sphere. Move the sphere somewhere that it is not initially colliding (making contact) with the box.
3. We will now tag the objects. Under sphere, see the element called tag, create a new tag. Call it ‘point10’. Make two more tags one which is ‘point100’ and then another called ‘point1000’. After creating point go back to the sphere and apply the tag ‘point10’, ‘point100’, or ‘point1000’. Unity will not do this for you automatically. We make three different ways of scoring in the case and you have creative license to change tag names and scoring. Refer to the script below to understand how this works.
4. Under the collider of the sphere (or any object that increases score when met/hit) check on is trigger. Note: once the object’s collider is set to ‘is trigger’ that same object’s rigidbody will become null (objects can pass through it and if it has gravity it can fall through objects). If you want an object (like an enemy) that you can bump into and triggers an event (‘is trigger’ is checked) the easiest fix is to create a child collider in a same/similar shape as the original collider except make it slightly bigger. Then check ‘is trigger’ on the bigger collider (the one which is the child of the object) and on the actual object have a collider that is not set to ‘is trigger’. Please keep in mind that you would have to tag the triggered collider.
5. Now create a new folder in Assets, call this Scripts.
6. Under scripts create a new C# file.
7. This code is the basic construct of how to use scoring. Place this in the file and make sure to save the file as ‘Scoring’ if you want to rename the C# file please refer to the Points for Consideration doc file.

|  |
| --- |
| /\*This program features a scoring method that is depedent on player collisions  The player must collide with specifically tagged colliders to aquire points  The points are visible on a UI Canvas which also features a countdown timer  UI elements for score and time left must be created and configured in Unity  For reference on how to use this script please consult the Scoring document  for any questions regarding how to use this script please contact [kai02916@gmail.com](mailto:kai02916@gmail.com)  Created by: Kathrine von Friedl for the University of Waterloo's Ideas Clinic\*/  using System.Collections; using System.Collections.Generic; using UnityEngine; using UnityEngine.UI;  public class Scoring : MonoBehaviour {  private float timeLeft = 60;   public GameObject timeLeftUI;  public GameObject scoreUI;  static int playerScore = 0;     void Start()  {  UpdateScore(playerScore);  }   void UpdateScore(int addedValue)  {  playerScore += addedValue;  scoreUI.gameObject.GetComponent<Text>().text = "Score: " + playerScore;  }   void OnTriggerEnter(Collider other)  {  if (timeLeft > 0)  {  switch (other.tag)  {  case "point10": //the name of the tag   UpdateScore(10); //calling another function with 10  break;  case "point100":  UpdateScore(100);  break;  case "point1000":  UpdateScore(1000);  Break;  }  }  }   void OnTriggerExit(Collider other)  {  Debug.Log("Object Exited the trigger");  }   void Update()  {  //end of new code   if (timeLeft > 0)  {  timeLeft -= Time.deltaTime;  timeLeftUI.gameObject.GetComponent<Text>().text = ("Time Left: " + (int)timeLeft);  }  else  timeLeftUI.gameObject.GetComponent<Text>().text = ("Time is up!");  } } |

1. Drag the scoring script onto the player
2. Create an empty game object, rename to UIElements (this is strictly for organizational purposes)
3. Underneath UIElements click create → UI → text (this will create a text UI element under a canvas UI element as well as an event system)
4. Duplicate the text, we need two text UI elements
5. Rename one text element to Score and another to TimeLeft
6. Under each text element, change the text under Text (Script) to “This Will Be Score” and “This Will Be Time” respectively. Note: the scripts we wrote will update the text of each element once per frame. The reason for you writing these sentences is for you to visually understand the place of each text box and nothing more. You don’t need text in either but that will make editing a lot harder unless you know what you are doing.
7. Under the player, go to the scoring script and drag and drop each UI text element (Score and TimeLeft) into the appropriate bins
8. The canvas and text will naturally be far TOO BIG for your project. We need to add some anchors and rescale sizes. Start by clicking on Canvas. Change ‘Render Mode’ under Canvas to ‘World Space’. Next, drag the CenterEyeAnchor from the hierarchy (pictured below) to the ‘Event Camera’ under Canvas.

|  |  |
| --- | --- |
|  |  |

1. Next we will scale down the text to hover at a point in space beyond where the user can move. Play around with the transforms of the UIElements gameObject as well as its children. You can also change the size of the text by increasing font size. An ideal setup features the largest font size. If this makes the text too big, simply change the transform on either UIElements, Canvas or the Rect Transform on the text to have a smaller scale. This increases clarity of the text, removing the fuzziness. Note: On various projects I have experienced a variety of max font sizes. By this I mean that occasionally a font size of 60 is fine. However, a font size of 28 may not show any text. The size incrementally increases from 1-27 but at 28 the text disappears.
2. Move the UIElements or Canvas to lie over the edge of the platform. You don't want to be able to walk past the text, look up and see it inverted.
3. With that you should be able to score! Drive the cube towards to sphere, when a collision occurs the sphere should increment the score. Note: the script prevents from points being scored after time is up.
4. Expanding on this tutorial, you probably do not want to increase points after walking through a physical object. We used a sphere for ease of the tutorial but this same principle can be applied to a hole in a basket (for basketball) or a hole in a door frame (for a doorway). Both feature the above steps applied on a collider without a mesh.