

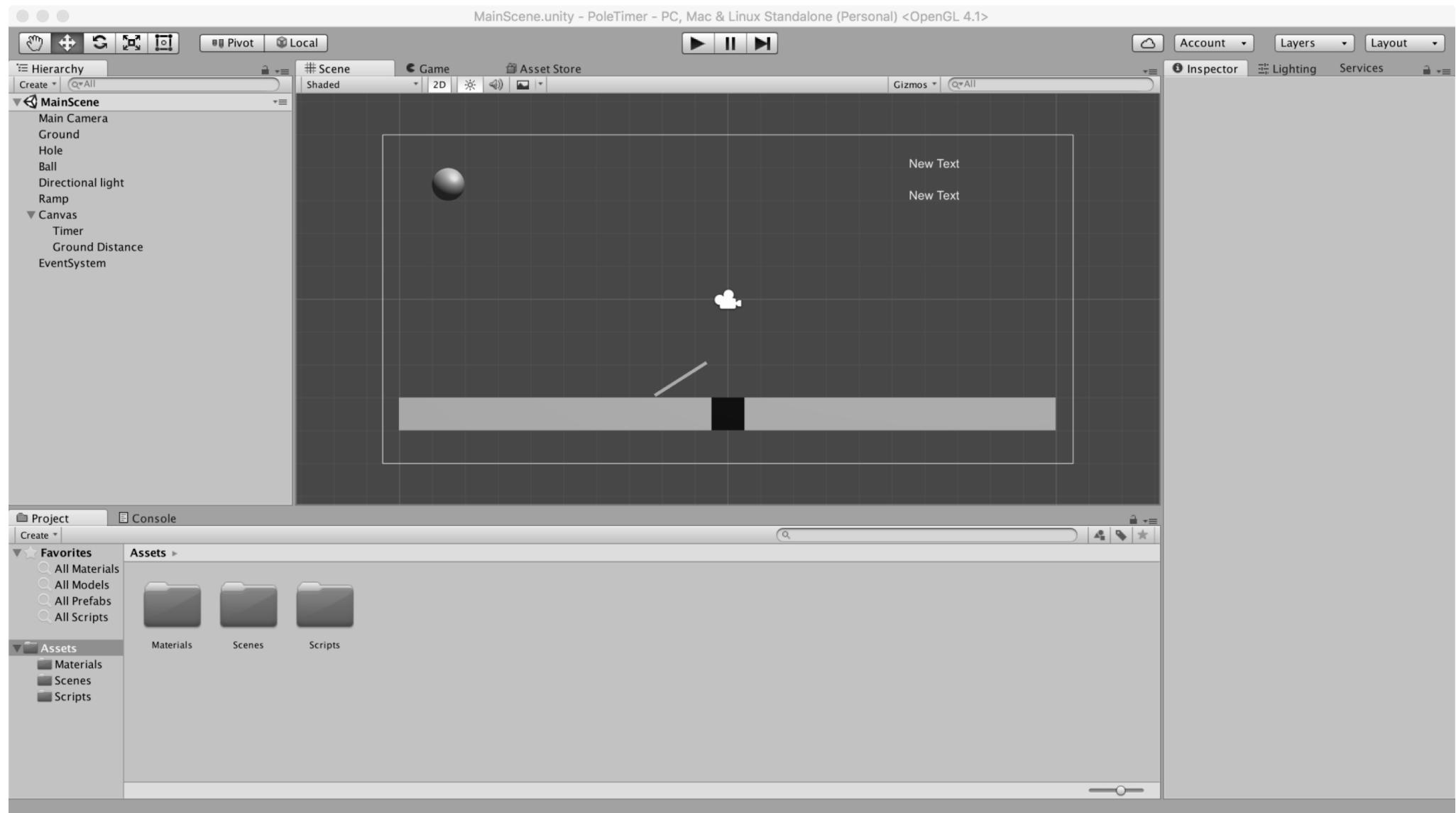
CSC476: Game Development

Class 14

October 11, 2018

Today

- Create the “Pole Timer Game” game.
- Unity Concepts: Persisting Data, RayCasting
 - Stop watch feature
- Deploying Games



“Pole Timer Game”

- No pole...
 1. ...but a timer
 2. Horizontal and Vertical RayCasting
 3. Persisting Data (“high scores”, etc.)
- More practice with UI and Adding Force
- *Game Objective:* game stops when ball has hovered over the “landing area” for 5 consecutive seconds. (can be touching ground or a little above it)

The Setup

- 2D Game; Save the default Scene as “MainScene”.
 - Remember to create a “Scripts” folder.
- Create a ground.
- Create a “hole” (landing area).
- Create a ball.
- Color and position the GameObjects.
- Add a directional light.
- Add a Rigidbody to the ball.

Adding Force to the Ball

```
public Rigidbody rb;
public float thrustX = 3.0f;
public float thrustY = 10.0f;

// Use this for initialization
void Start () {
    rb = GetComponent<Rigidbody>();
}

// Update is called once per frame
void Update () {
    // Debug.Log("Update time :" + Time.deltaTime);
}

void FixedUpdate () {
    // Debug.Log("FixedUpdate time :" + Time.deltaTime);

    // The value will be in the range -1...1
    rb.AddForce(Input.GetAxis("Horizontal") * thrustX,0,0);
    rb.AddForce(0,Input.GetAxis("Vertical") * thrustY,0);
}
```

Update() vs. FixedUpdate()

- `Update()`:
 - Called every frame
 - Typically used for *regular updates*, such as: moving non-physics objects, timers, receiving input, ...
 - Interval timing may be inconsistent (if something takes a while to process)
- `FixedUpdate()`:
 - Similar to `Update`, except Unity will try to call `FixedUpdate` at a consistent, regular time interval
 - Typically used for *physics updates*, such as moving Rigidbody objects

More Setup

- Create a little ramp before the ground.
- Create a UI Text Object for the Timer.

Simulating a Stop Watch

```
using UnityEngine.UI;
```

```
private double time;

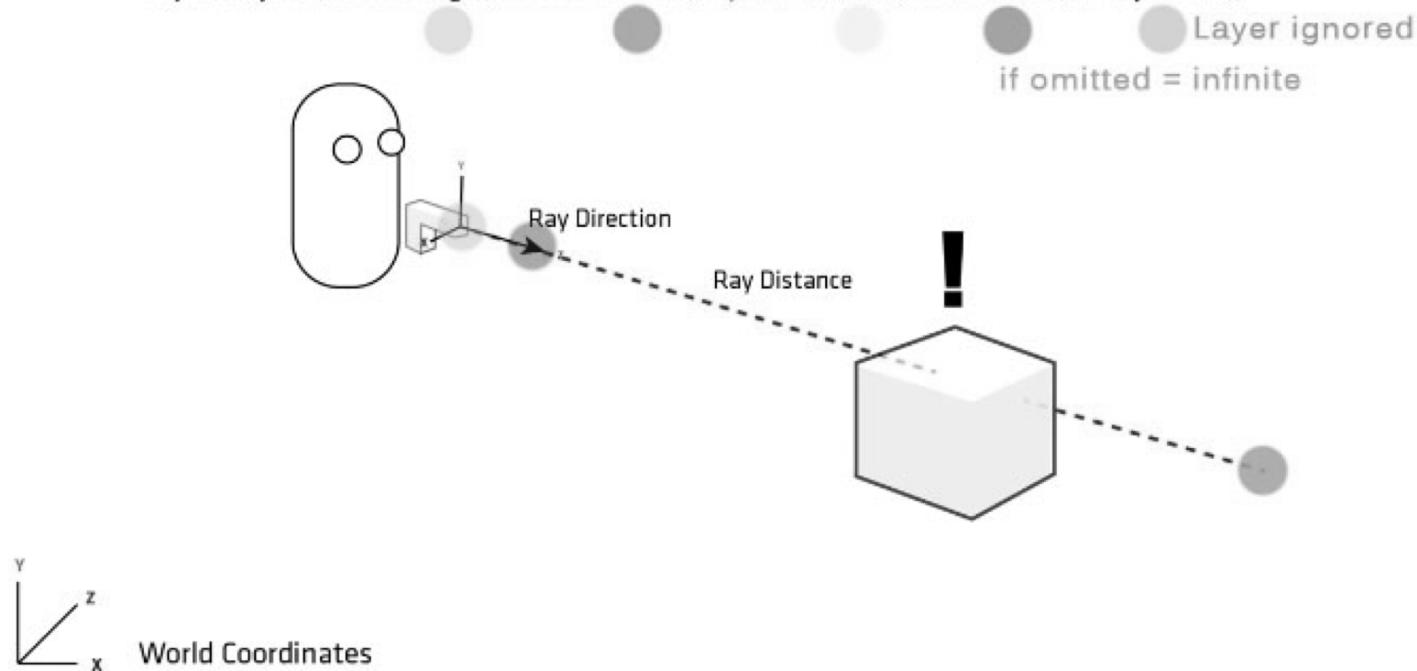
public Text timer;

// Update is called once per frame
void Update () {
    // Debug.Log("Update time :" + Time.deltaTime);

    time += Time.deltaTime;
    timer.text = time.ToString();
}
```

RayCasting

```
Physics.Raycast(Vector3 origin, Vector3 direction, RaycastHit hitInfo, float distance, int LayerMask);
```



More Setup...

- Another UI Text Object

How Close is the Ball to the Ground?

```
public Text GroundReport;

// Update is called once per frame
void Update () {

    RaycastHit hitInfo;
    if (Physics.Raycast(transform.position, Vector3.down, out hitInfo, 3.0f))
    {
        GroundReport.text = hitInfo.distance.ToString();
    }
}
```

out keyword causes arguments to be passed by reference:
<https://msdn.microsoft.com/en-us/library/t3c3bfhx.aspx>

Debugging RayCast

- RayCasts are invisible
- Can sometimes be tough to debug
- Option to view them in the scene view:

```
Debug.DrawRay(eyeLine.position, Vector2.right, Color.magenta, 0.1f);
```

<https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>

Persisting Data: PlayerPrefs

- Unity stores key/value pairs in the local computer's...
 - ...registry (Windows)
 - ...properties (Mac)

```
PlayerPrefs.GetFloat ("besttime", 100f);
```

- Return the value 100 if the “besttime” key is not found.

```
PlayerPrefs.SetFloat ("besttime", (float)winningTime);
```

```
private double winningTime;
```

Keeping Track of the Fastest Time

```
RaycastHit hitInfo;
if (Physics.Raycast(transform.position, Vector3.down, out hitInfo, 3.0f))
{
    GroundReport.text = hitInfo.distance.ToString();

    if (hitInfo.collider.tag == "Finish") {
        GroundReport.text = "HOLD...";
        winningTime += Time.deltaTime;
    } else {
        winningTime = 0.0;
    }

    if (winningTime > 5.0) {
        Time.timeScale = 0.0f; // "pauses" game
        // https://docs.unity3d.com/ScriptReference/Time-timeScale.html

        // https://docs.unity3d.com/ScriptReference/PlayerPrefs.html
        if (PlayerPrefs.GetFloat ("besttime", 100f) > winningTime) {
            PlayerPrefs.SetFloat ("besttime", (float)winningTime);
            Debug.Log ("New best time: " +
            PlayerPrefs.GetFloat ("besttime", 0f));
        }
    }
}
```

Deploying Games

- File -> “Build Settings”

GitHub for Unity

- <https://unity.github.com/>