Learning Journal

Assignment 1-

For my first tutorial I have created a code that allows an object to continuously rotate when the left key is selected.

Learning plan-

Task: Create and understand a code that allows objects to rotate in unity, when a key is pressed.

Step 1- Gain more knowledge of if statements and of input actions to create code

Step 2- Create code that allows object to rotate

Step 3- Create an object to implement code too

Step 4- Using unity software, attach code to object and run, to ensure code works well

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Time started | Time finished | Length of any distraction | Time taken to complete |
| Step 1 | 22/10/18  10.30am | 22/10/18  11.05am | 5 Minutes | 30 minutes |
| Step 2 | 22/10/18  11.30am | 22/10/18  11.50am | NONE | 20 Minutes |
| Step 3 | 23/10/18  2.35pm | 23/10/18  2.50pm | 4 Minutes | 11 Minutes |
| Step 4 | 24/10/18  9.30pm | 24/10/18  10pm | NONE | 30 Minutes |

Self assessment:

Overall I feel I have a clear understanding of how programming works and the main function of each unique input in code. However I wanted to research and gain knowledge of anything within the code that I was not completely sure on, or if this was the easiest and simplest code I could use to create a rotation.

Resources:

I will be using the official unity website in order to gain more knowledge of if statements. I already have clear knowledge on how to add an input key function.

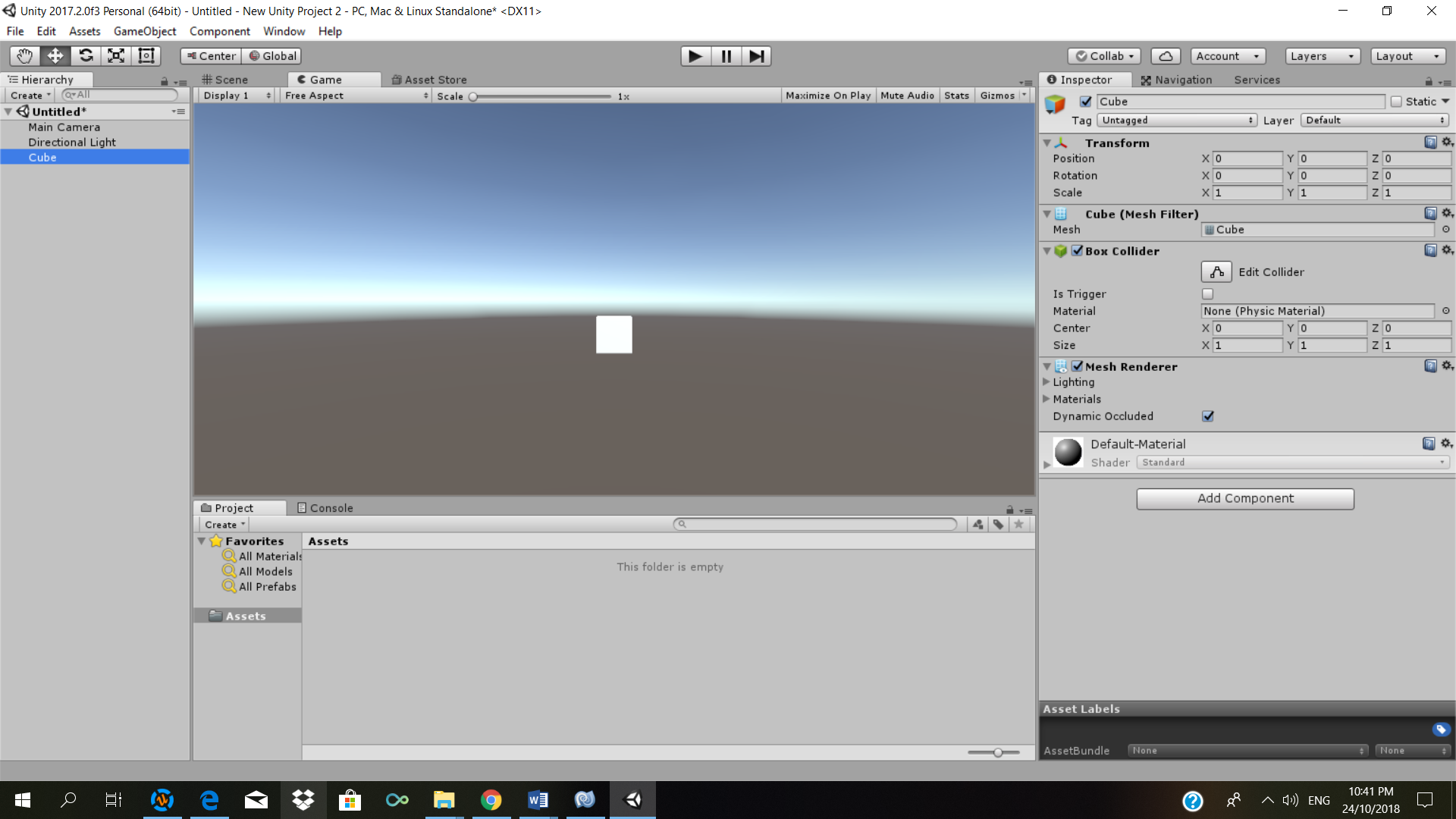
Link:

<https://unity3d.com/learn/tutorials/topics/scripting/if-statements>

<https://docs.unity3d.com/ScriptReference/Input.GetKeyDown.html>

**The object**

Here is the object I created in Unity to show how the code effects object movement.



**Code**

*using System.Collections;*

*using System.Collections.Generic;*

*using UnityEngine;*

*public class Rotate : MonoBehaviour {*

*// Use this for initialization*

*void Start () {*

*}*

*void Update () {*

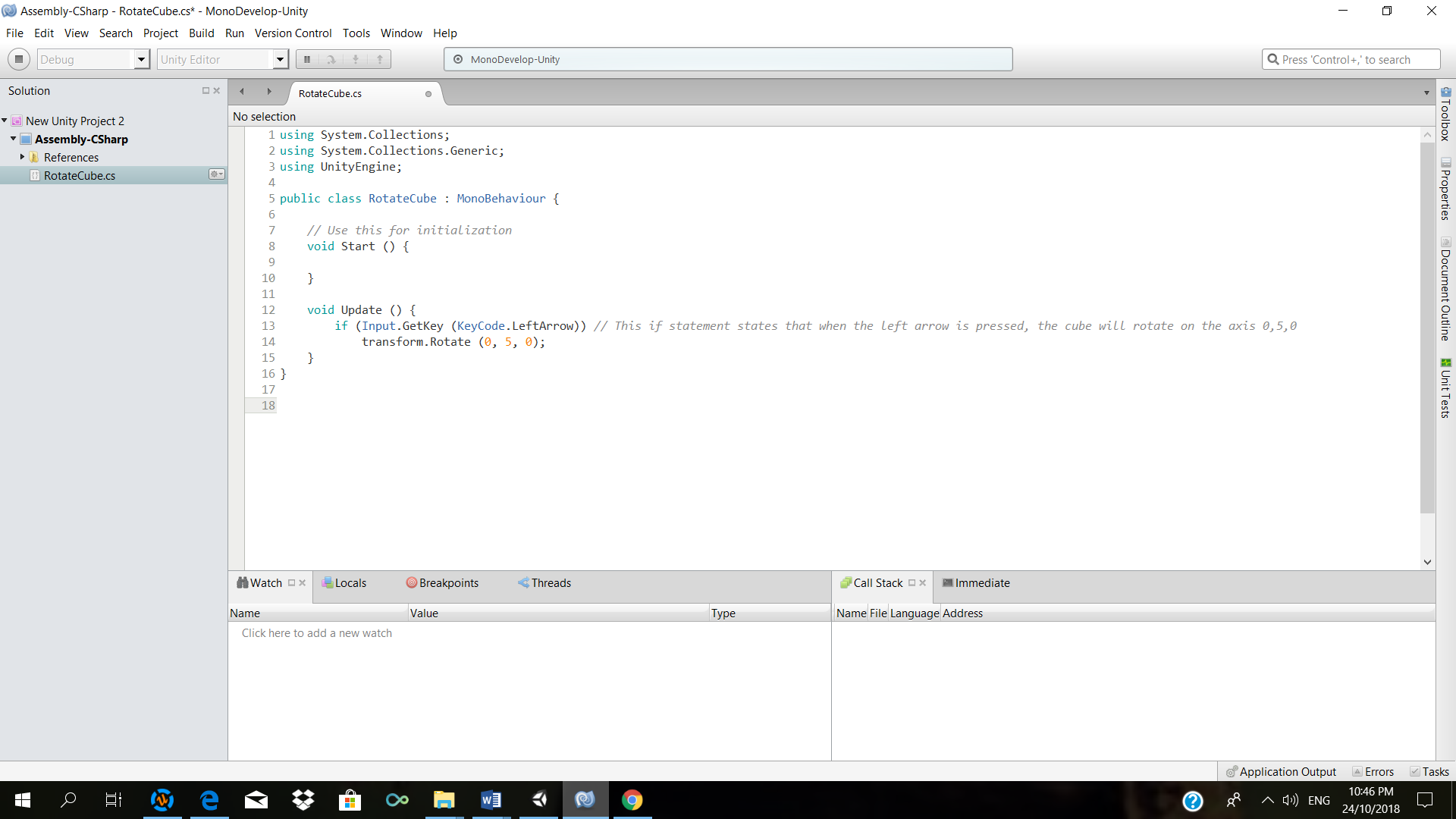
*if (Input.GetKey (KeyCode.LeftArrow))*

*transform.Rotate (0, 5, 0);*

*}*

*}*

Here is the code as shown in the unity software:



Learning Plan 2:

With the tutorials I want to make the code slightly more complex as the tutorials go on.

Learning plan- In this tutorial I want to make a moving object, moving left, right, up and down that can be destroyed when it collides with another object.

Task 1- Gain knowledge on how to create movement within a 3D object in unity, when selecting the left,right, up and down arrow keys.

Task 2- Learn how to destroy an object when it collides with another object, using tags and scripting.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Time started | Time finished | Length of any distraction | Time taken to complete |
| Step 1 | 26/10/18  10.30am | 07/10/18 | ----- | 2 Minutes |
| Step 2 | 27/10/18  11.30am | 27/10/18  2pm | 30 minutes | 2.5 hours |
| Step 3 | 28/10/18  2.35pm | 28/10/18  2. 38pm | ------- | 2 minutes |
| Step 4 | 28/10/18  3pm | 28/10/18  3.10pm | 10 minutes | 1 hour |
| Step 5 | 28/10/18  3.10pm | 28/10/12  3.15pm | ------ | 5 minutes |
| Step 6 | 28/10/18  3.17pm | 28/10/18  3.19pm | ------- | 2 minutes |
| Step 7 | 28/10/18  5pm | 28/10/18  5.35pm | 15 minutes | 35 minutes |

Step 1- The first step is to create the object you wish to move, by selecting GameObject, 3D Object and the object you wish to create, you can also add a material in order to give the object some colour. Select Assets,Material, select a colour and add it to the chosen shape.

Step 2- Create code that allows object to move-

*using System.Collections;*

*using System.Collections.Generic;*

*using UnityEngine;*

*public class Movement : MonoBehaviour {*

*public float moveSpeed = 3f; //this is the speed in which the cube will travel*

*void Start () {*

*}*

*void Update ()*

*{ //this allows the cube to move up and down*

*transform.Translate(Vector3.forward \* Time.deltaTime \* Input.GetAxis("Vertical")\* moveSpeed);*

*//this allows the cube to let and right*

*transform.Translate(Vector3.right \* Time.deltaTime \* Input.GetAxis("Horizontal")\* moveSpeed);*

*}*

*}*

Step 3 - The next step is to now create the object the moving player will collide with, so again, make another 3D object.

Step 4- You now need to create the code to cause the collision and destroying the object-

*public class Destroy : MonoBehaviour*

*{*

*void OnTriggerEnter(Collider col)*

*{*

*if (col.gameObject.CompareTag("Crash"))*

*{*

*Destroy(gameObject);*

*}*

*}*

*}*

Step 5- once the code has been created, you will need to add a tag to the enemy sprite ( the one who causes the other object to be destroyed), this will need to be the same as referenced in the code. I used the tag ‘Crash’.

Step 6- Ensure the Collision on the enemy object is set to ‘is trigger’ in the inspector section.

Step 7- The last step is to add a code to the enemy sprite to continuously move left and right-

*using System.Collections;*

*using System.Collections.Generic;*

*using UnityEngine;*

*public class LeftandRight : MonoBehaviour {*

*public float delta = 1.5f; // Amount to move left and right from the start point*

*public float speed = 2.0f; //the speed in which it moves*

*private Vector3 startPos;*

*void Start()*

*{*

*startPos = transform.position;*

*}*

*void Update()*

*{*

*Vector3 v = startPos;*

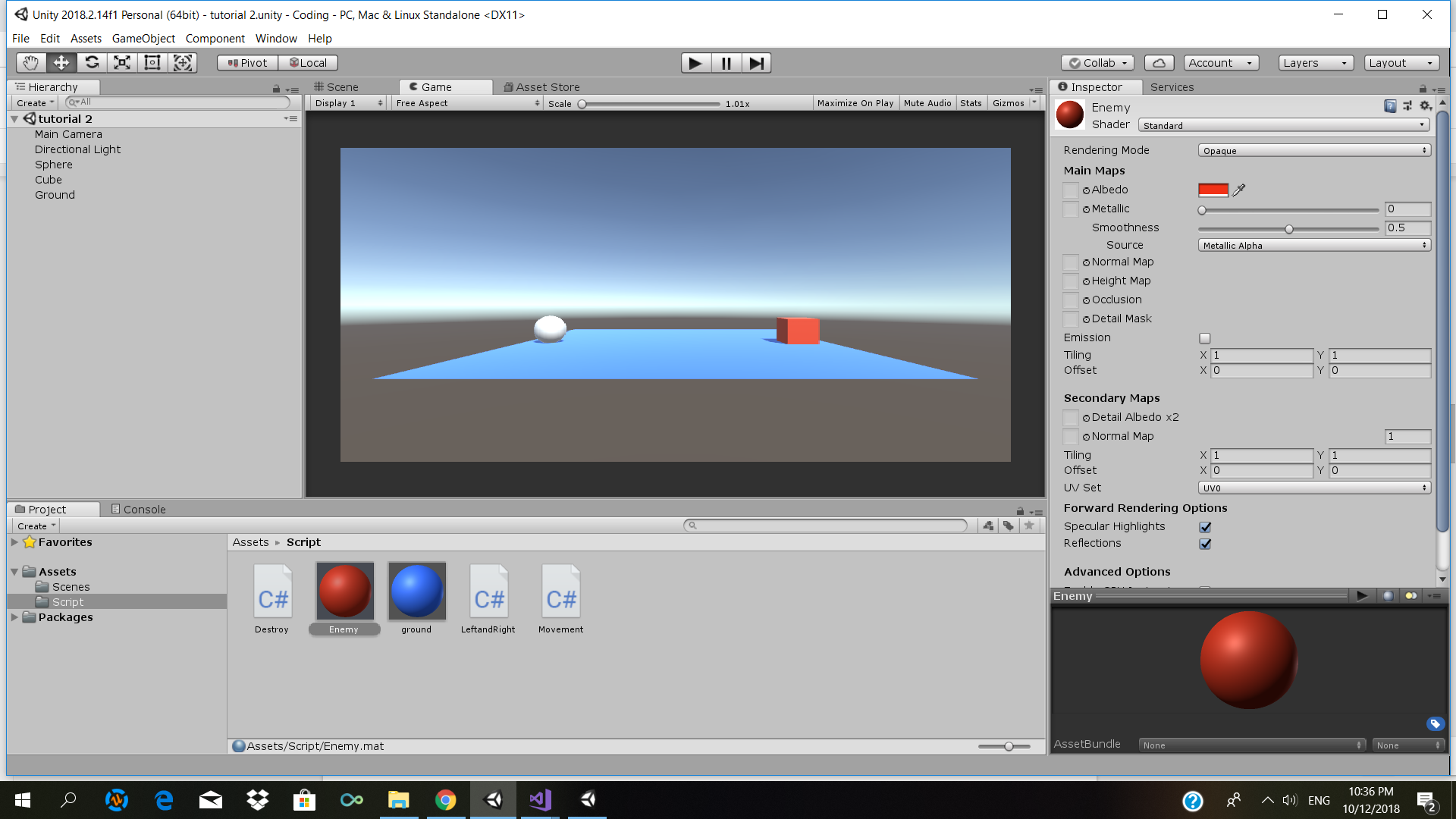
*v.x += delta \* Mathf.Sin(Time.time \* speed); //what direction the object should move and how fast*

*transform.position = v;*

*}*

*}*

This is what the scene looked like, including the objects I created-



Resources:

In order to create this tutorial I had to research how to destroy an object, make an object move on its own repeatedly and also how to make the objects move in all directions.

<https://unity3d.com/learn/tutorials/topics/scripting/destroy>

<https://gamedev.stackexchange.com/questions/129212/moving-a-platform-left-to-right-with-continuous-movement>

Tutorial 3

In thi tutorial i will be creating a script that allows the player to be able to see which enemy is closest to it, using a line. This will help me in future combate games.

Task 1- research how to be able to identify what object is closest to the player.

Task 2- be able to create a line to show distance in a unity game.

Step 1- Create 3 objects, 2 of the same shape and one unique one. The unique shape will work as the main player. The other 2 shapes will work as the enemy.

Step 2- Create a movement script for the player, this will allow us to see if the script we are creating works-

*using System.Collections;*

*using System.Collections.Generic;*

*using UnityEngine;*

*public class Movement : MonoBehaviour {*

*public float moveSpeed = 3f; //this is the speed in which the cube will travel*

*void Start () {*

*}*

*void Update ()*

*{ //this allows the cube to move up and down*

*transform.Translate(Vector3.forward \* Time.deltaTime \* Input.GetAxis("Vertical")\* moveSpeed);*

*//this allows the cube to let and right*

*transform.Translate(Vector3.right \* Time.deltaTime \* Input.GetAxis("Horizontal")\* moveSpeed);*

*}*

*}*

Step 3 - Create the code that shows the distance between a player and the closest enemy and produces the answer with a line-

*using System.Collections;*

*using System.Collections.Generic;*

*using UnityEngine;*

*public class FindClosest : MonoBehaviour*

*{*

*// Update is called once per frame*

*void Update()*

*{*

*FindClosestEnemy();*

*}*

*void FindClosestEnemy()*

*{*

*float distanceToClosestEnemy = Mathf.Infinity;*

*Enemy closestEnemy = null;*

*Enemy[] allEnemies = GameObject.FindObjectsOfType<Enemy>();*

*foreach (Enemy currentEnemy in allEnemies)*

*{*

*float distanceToEnemy = (currentEnemy.transform.position - this.transform.position).sqrMagnitude;*

*if (distanceToEnemy < distanceToClosestEnemy)*

*{*

*distanceToClosestEnemy = distanceToEnemy;*

*closestEnemy = currentEnemy;*

*}*

*}*

*Debug.DrawLine(this.transform.position, closestEnemy.transform.position);*

*}*

*}*

Step 4- Create a script, named enemy and attach it to each enemy object. This will help refer to the enemy object in our ‘FindClosest Script’.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Time started | Time finished | Length of any distraction | Time taken to complete |
| Step 1 | 11/11/18  3.45pm | 11/11/18  3.55pm | ---- | 10 mins |
| m | 11/11/18  4pm | 11/11/18  4.05pm | --- | 3 minutes  (previously used in task 2) |
| Step 3 | 11/11/18  4.10pm | 2/11/18  5.30pm | 10mins | 1.2 hours |
| Step 4 | 11/11/18  6.05pm | 11/11/18  6.10pm | ----- | 5 minutes |

Tutorial 4

In this tutorial I will be creating a script that changes sprite colours once the game begins to play. This could become handy for testing games in the future, as well as help me add materials or effects during the game.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Time started | Time finished | Length of any distraction | Time taken to complete |
| Step 1 | 2/11/18  2pm | 2/11/18  2.05pm | --- | 3 minutes |
| Step 2 | 2/11/18  2.10pm | 2/11/18  3pm | 3 minutes | 48 minutes |
| Step 3 | 2/11/18  3.05pm | 2/11/18  3.10pm | ----- | 5 minutes |

Step 1- Create the GameObjects that you wish to use, you can do as many or as minimal as you want.

Step 2- Create the code that will change the sprites material when the game begins to run-

*using System.Collections;*

*using System.Collections.Generic;*

*using UnityEngine;*

*public class ChangeSprite : MonoBehaviour {*

*private Renderer rend;*

*[SerializeField]*

*private Color colorToTurnTo = Color.white;*

*private void Start()*

*{*

*// Assign Renderer component to rend variable*

*rend = GetComponent<Renderer>();*

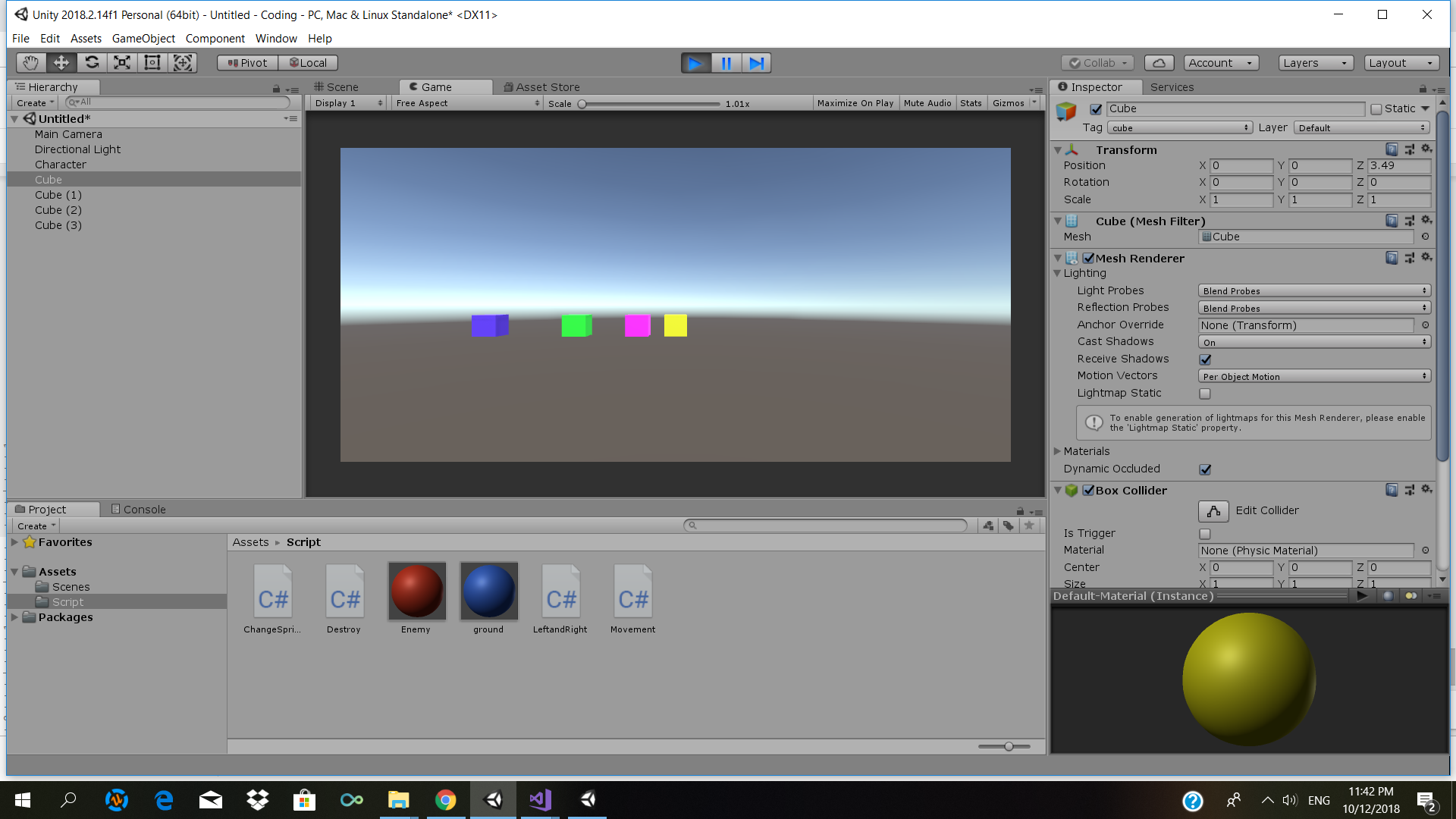
*// Change sprite color to selected color*

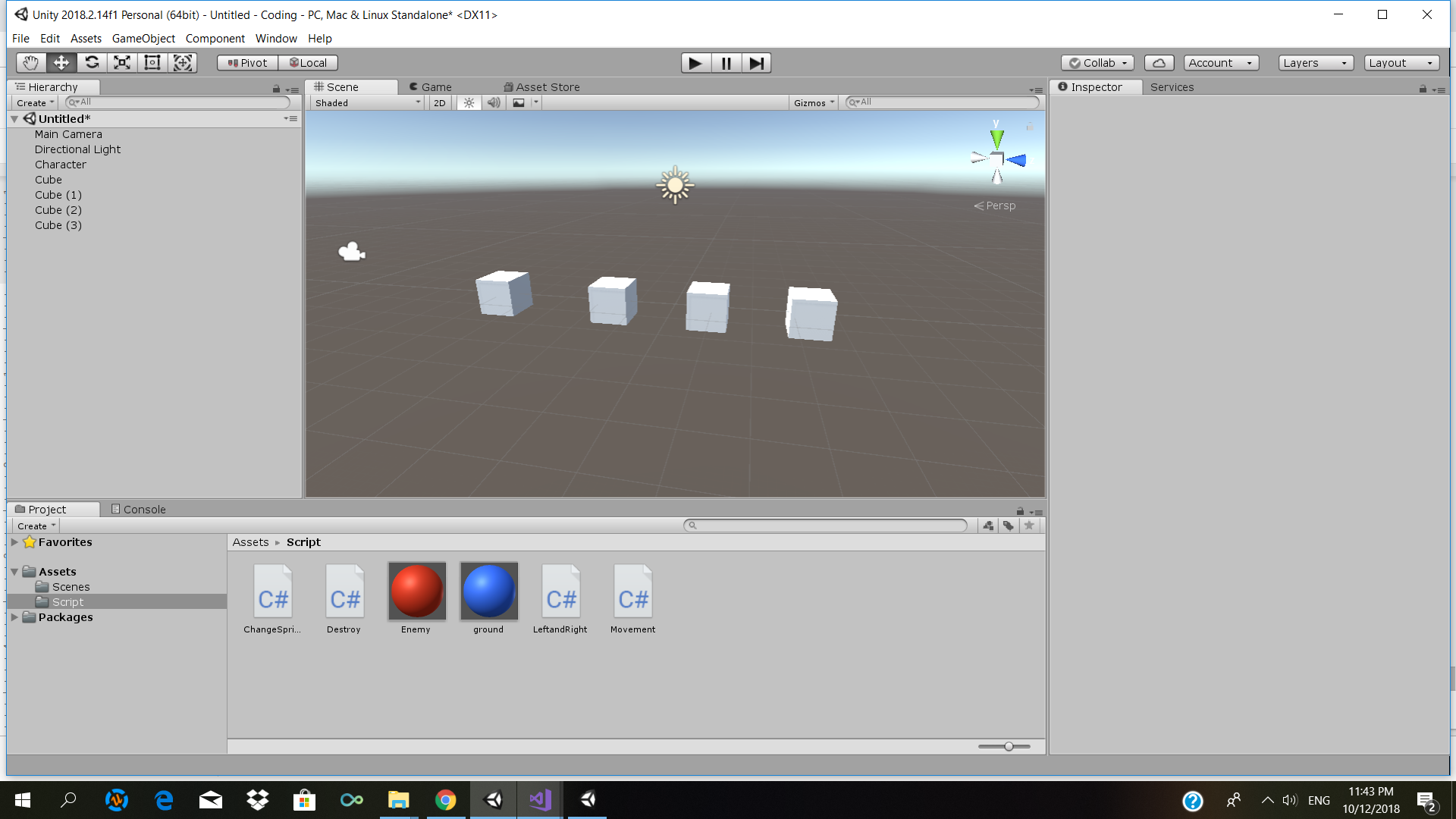
*rend.material.color = colorToTurnTo;*

*}*

*}*

Step 3- Drag the code on to each object and select whatever colours you would like,choice what colour you wish the object to turn into once the game will begin.





Before After