**LAB Manual 20**

**Rider**

* Camera Controller Script
* Rotate the Car

**Now we want that our camera fallow around our car**

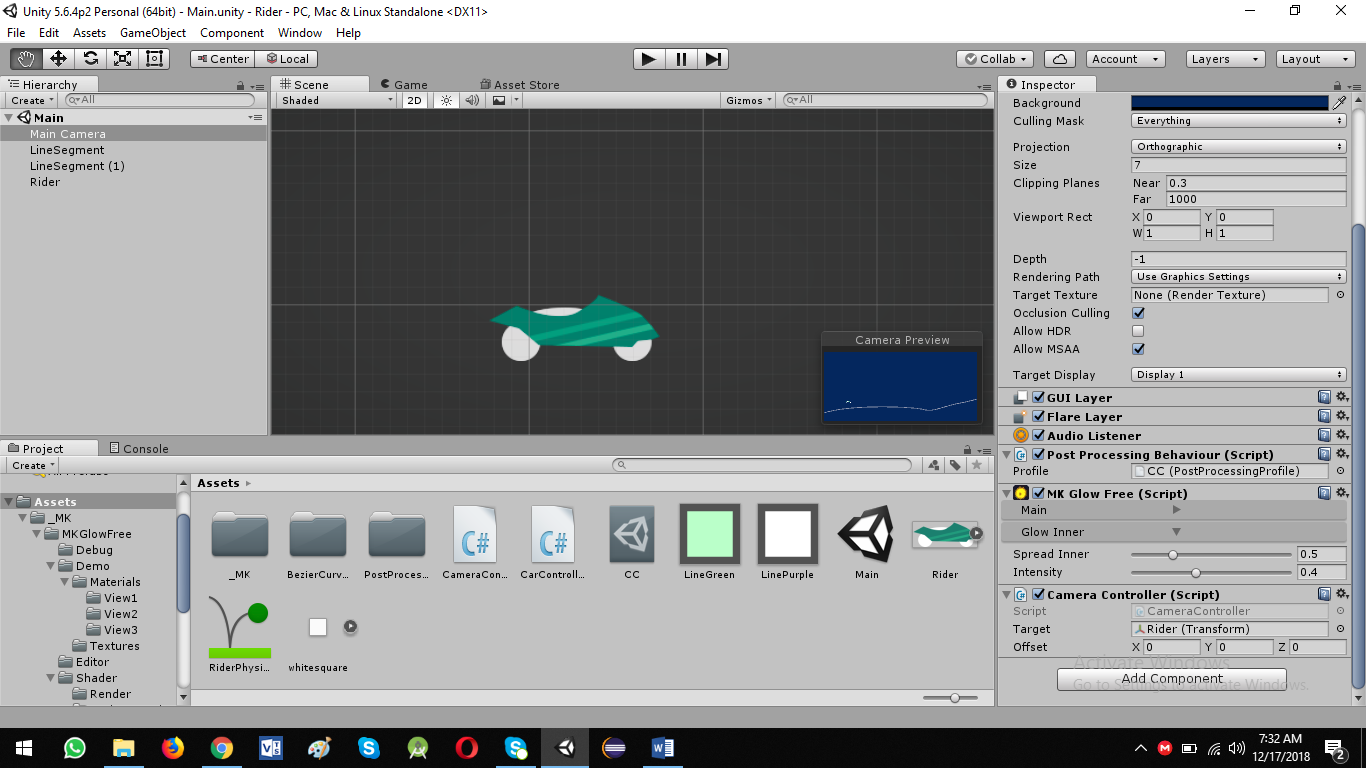
* Select Main Camera
* Add new component: CameraController script
* **LateUpdate():** Called at the end of drawing frame

**CameraController.cs**

using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class CameraController : MonoBehaviour {  
  
  
    public Transform target;  
    public Vector3 offset;  
  
    private void LateUpdate(){  
        Vector3 newPos= target.position + offset;  
        newPos.z = transform.position.z;  
        transform.position = newPos;  
  
      
    }  
}

Note:

Main Camera: Camera Controller(Script) -> Target -> drag Rider



**Rotate the Rider:**

Check if rider is on the ground/platform or touching the line.

If rider is in the air rotate it.

Let’s update the CarController Script.

private bool isGrounded = false;

private void OnCollisionEnter2D(){  
        isGrounded = true;  
    }  
  
    private void OnCollisionExit2D(){  
        isGrounded = false;  
          
    }

Now updated version is:

using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class CarController : MonoBehaviour {  
  
    private bool move = false;  
    private bool isGrounded = false;  
  
    public Rigidbody2D rb;  
  
    public float speed = 20f;  
  
    private void Update(){  
        if (Input.GetButtonDown ("Fire1")) { *// Mouse left click button pressed*  
            move = true;  
        }  
        if (Input.GetButtonUp ("Fire1")) { *// Mouse left click button released*  
            move = false;  
        }  
    }  
  
    private void FixedUpdate(){  
        if (move == true) {  
            rb.AddForce ( transform.right \* speed \* Time.fixedDeltaTime \* 100f , ForceMode2D.Force);  
              
        }  
    }  
  
  
    private void OnCollisionEnter2D(){  
        isGrounded = true;  
    }  
  
    private void OnCollisionExit2D(){  
        isGrounded = false;  
          
    }  
          
}

**Debug mode**

Click on Rider and check new variable isGrounded.

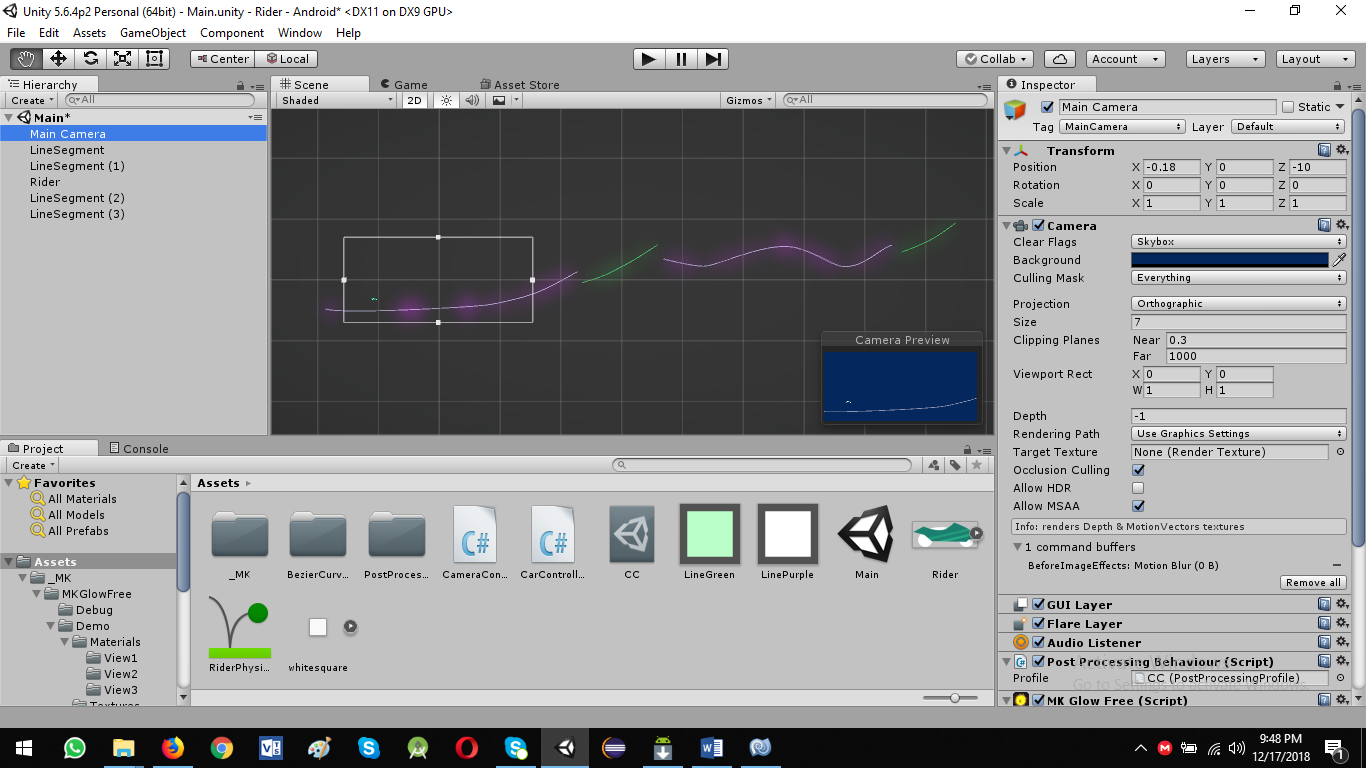
Now add the rotating code to CarController Script

public float rotationSpeed = 2f;

    private void FixedUpdate(){  
        if (move == true) {  
  
            if (isGrounded) {  
                rb.AddForce ( transform.right \* speed \* Time.fixedDeltaTime \* 100f , ForceMode2D.Force);  
            } else {  
                rb.AddTorque (rotationSpeed \* rotationSpeed \* Time.fixedDeltaTime \* 100f, ForceMode2D.Force);  
            }  
        }  
    }

Here is final version of CarController

using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class CarController : MonoBehaviour {  
  
    private bool move = false;  
    private bool isGrounded = false;  
  
    public Rigidbody2D rb;  
  
    public float speed = 20f;  
  
    public float rotationSpeed = 2f;  
  
    private void Update(){  
        if (Input.GetButtonDown ("Fire1")) { *// Mouse left click button pressed*  
            move = true;  
        }  
        if (Input.GetButtonUp ("Fire1")) { *// Mouse left click button released*  
            move = false;  
        }  
    }  
  
    private void FixedUpdate(){  
        if (move == true) {  
  
            if (isGrounded) {  
                rb.AddForce ( transform.right \* speed \* Time.fixedDeltaTime \* 100f , ForceMode2D.Force);  
            } else {  
                rb.AddTorque (rotationSpeed \* rotationSpeed \* Time.fixedDeltaTime \* 100f, ForceMode2D.Force);  
            }  
        }  
    }  
  
  
    private void OnCollisionEnter2D(){  
        isGrounded = true;  
    }  
  
    private void OnCollisionExit2D(){  
        isGrounded = false;  
          
    }  
          
}



Physics Enable on LineSegment3

* Rotate around its center
* When rider jumps over it

Add Component Rigidbody 2D to LineSegment to rotate the linesegment

And make some necessary settings

Mass: 100

Angular Drag: 100

Constraints: check x and y only