**LAPORAN MOBILE GRAPHICS AND GAME**



**Disusun oleh**

**M Aldhi N I - 145410100**

**Jurusan : TI / S1**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**A K A K O M**

**YOGYAKARTA**

**2019**

url assets: <https://assetstore.unity.com/packages/3d/vehicles/roadster-low-poly-72196>

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class rotasi : MonoBehaviour {

float rotSpeed = 20;**->untuk kecepatan rotasi**

void OnMouseDrag()

{

float rotX = Input.GetAxis("Mouse X")\*rotSpeed\*Mathf.Deg2Rad;

float rotY = Input.GetAxis("Mouse Y")\*rotSpeed\*Mathf.Deg2Rad;

transform.RotateAround(Vector3.up, -rotX);

transform.RotateAround(Vector3.right, rotY);

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PinchZoom : MonoBehaviour {

public float perspectiveZoomSpeed = 0.5f; // The rate of change of the field of view in perspective mode.

public float orthoZoomSpeed = 0.5f; // The rate of change of the orthographic size in orthographic mode.

void Update()

{

// If there are two touches on the device...

if (Input.touchCount == 2)

{

// Store both touches.

Touch touchZero = Input.GetTouch(0);

Touch touchOne = Input.GetTouch(1);

// Find the position in the previous frame of each touch.

Vector2 touchZeroPrevPos = touchZero.position - touchZero.deltaPosition;

Vector2 touchOnePrevPos = touchOne.position - touchOne.deltaPosition;

// Find the magnitude of the vector (the distance) between the touches in each frame.

float prevTouchDeltaMag = (touchZeroPrevPos - touchOnePrevPos).magnitude;

float touchDeltaMag = (touchZero.position - touchOne.position).magnitude;

// Find the difference in the distances between each frame.

float deltaMagnitudeDiff = prevTouchDeltaMag - touchDeltaMag;

// If the camera is orthographic...

if (GetComponent<Camera>().orthographic)

{

// ... change the orthographic size based on the change in distance between the touches.

Camera.main.orthographicSize += deltaMagnitudeDiff \* orthoZoomSpeed;

// Make sure the orthographic size never drops below zero.

Camera.main.orthographicSize = Mathf.Max(GetComponent<Camera>().orthographicSize, 0.1f);

}

else

{

// Otherwise change the field of view based on the change in distance between the touches.

Camera.main.fieldOfView += deltaMagnitudeDiff \* perspectiveZoomSpeed;

// Clamp the field of view to make sure it's between 0 and 180.

Camera.main.fieldOfView = Mathf.Clamp(GetComponent<Camera>().fieldOfView, 0.1f, 179.9f);

}

}

}

}