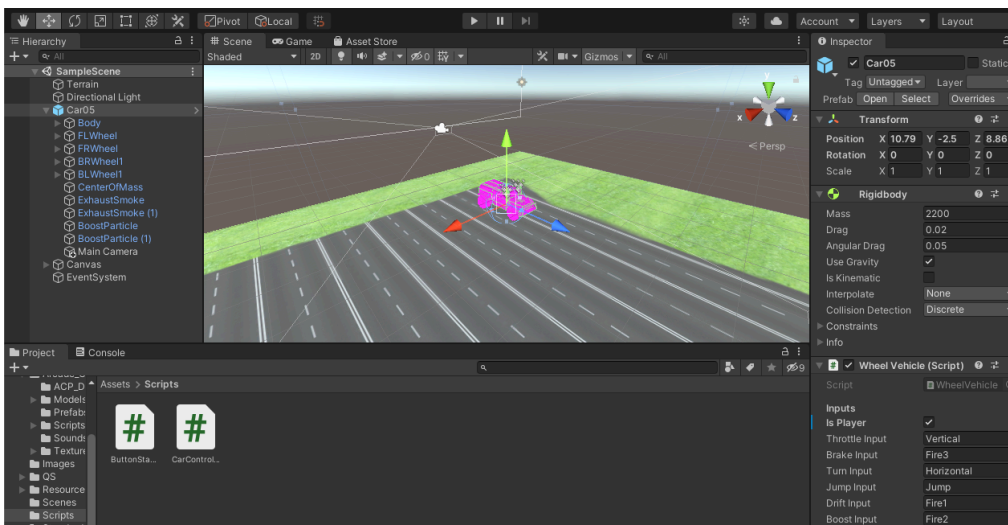


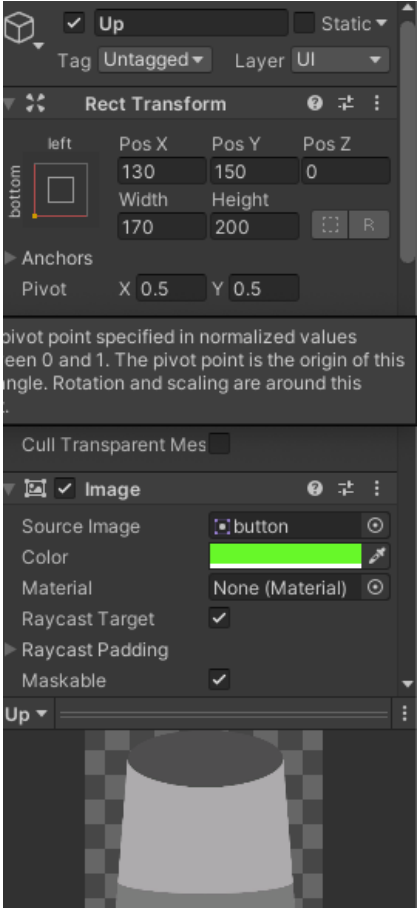
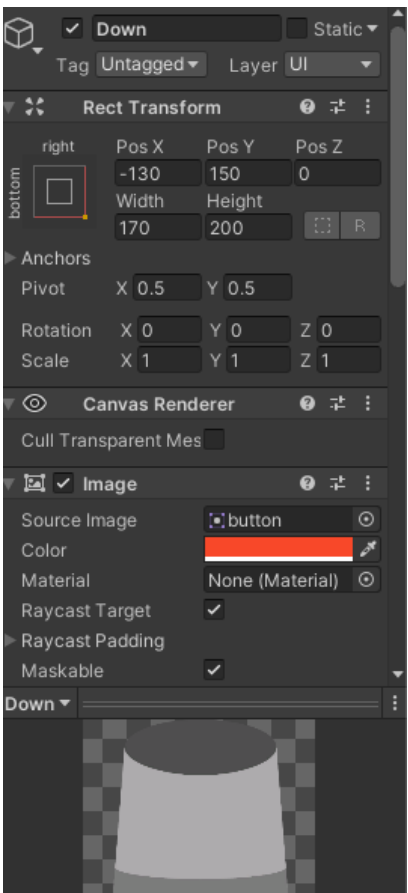


Jurusan Teknologi Informasi Politeknik Negeri Malang
Tugas Minggu: UTS
Mata Kuliah Komputasi Multimedia
Pengampu:
Februari 2021

Nama : Muhammad Iqbaluddin Al Huda
Kelas : TI-3E
NIM : 1841720013

Langkah	Keterangan
1	
2	Inspector memasukkan script wheel vehicle

	Inspector memasukkan scipt Engine Sound Manager
3	
4	Button up

	 <p>The screenshot shows the Unity Inspector for a UI element named 'Up'. It is a 'Rect Transform' with a 'bottom' anchor. The position is (130, 150, 0) with a width of 170 and height of 200. The pivot is at (0.5, 0.5). The 'Image' component has a source image of 'button' and a color of bright green. The 'Canvas Renderer' component is also visible.</p>
5	<p>Button down</p>  <p>The screenshot shows the Unity Inspector for a UI element named 'Down'. It is a 'Rect Transform' with a 'bottom' anchor. The position is (-130, 150, 0) with a width of 170 and height of 200. The pivot is at (0.5, 0.5). The 'Image' component has a source image of 'button' and a color of bright orange. The 'Canvas Renderer' component is also visible.</p>
6	<p>Class ButtonStateManager</p>

```

ButtonStateManager.cs x CarController.cs CameraFollow.cs EngineSoundManager.cs
Users > md1010 > Documents > Semester 6 > Komputasi Multimedia Project > UTS > UTS > Assets > Scripts
1 using UnityEngine.EventSystems;
2
3 public class ButtonStateManager : EventTrigger
4 {
5     public bool downed = false;
6
7
8     public override void OnPointerDown(PointerEventData data)
9     {
10         downed = true;
11     }
12
13     public override void OnPointerUp(PointerEventData data)
14     {
15         downed = false;
16     }
17
18 }
19

```

Class CarController

```

ButtonStateManager.cs x CarController.cs x CameraFollow.cs EngineSoundManager.cs
Users > md1010 > Documents > Semester 6 > Komputasi Multimedia Project > UTS > UTS > Assets > Scripts > CarController.cs
6 public class CarController : MonoBehaviour
7 {
8
9     public GameObject car;
10    public GameObject up;
11    public GameObject down;
12
13    private WheelVehicle carComponent;
14
15    void Start()
16    {
17        carComponent = car.GetComponent<WheelVehicle>();
18    }
19
20    void Update()
21    {
22
23
24        if(up.GetComponent<ButtonStateManager>().downed) {
25            if(carComponent.manualityThrottle < 1f)
26                carComponent.manualityThrottle += 0.1f;
27        }
28        else if(down.GetComponent<ButtonStateManager>().downed) {
29            if(carComponent.manualityThrottle > -1f)
30                carComponent.manualityThrottle -= 0.1f;
31        }
32        else{
33            if(carComponent.manualityThrottle != 0f) {
34
35                if(carComponent.manualityThrottle < 0f)
36                    carComponent.manualityThrottle += 0.05f;
37
38                if(carComponent.manualityThrottle > 0)
39                    carComponent.manualityThrottle -= 0.05f;
40            }
41        }
42    }
43
44 }

```

7

8

Class CameraFollow

```

5 namespace VehicleBehaviour.Utils {
6     public class CameraFollow : MonoBehaviour {
7         [SerializeField] bool follow = false;
8         public bool Follow { get { return follow; } set { follow = value; } }
9         [SerializeField] Transform target;
10        [SerializeField] Transform[] targets;
11        [SerializeField] Vector3 offset;
12        [Range(0, 10)]
13        [SerializeField] float lerpPositionMultiplier = 1f;
14        [Range(0, 10)]
15        [SerializeField] float lerpRotationMultiplier = 1f;
16
17        Rigidbody rb;
18
19        void Start () {
20            rb = GetComponent<Rigidbody>();
21        }
22
23        public void SetTargetIndex(int i) {
24            target = targets[i % targets.Length];
25        }
26
27        void FixedUpdate() {
28            if (!follow) return;
29
30            this.rb.velocity.Normalize();
31
32            Quaternion curRot = transform.rotation;
33
34            Rigidbody rb = target.GetComponent<Rigidbody>();
35            if (rb == null)
36                transform.LookAt(target);
37            else {
38                transform.LookAt(target.position * + target.forward * rb.velocity.sqrMagnitude);
39            }

```

Class EngineSoundManager

```

5 namespace VehicleBehaviour {
6     [RequireComponent(typeof(WheelVehicle))]
7     [RequireComponent(typeof(AudioSource))]
8
9     public class EngineSoundManager : MonoBehaviour {
10
11        [Header("AudioClips")]
12        public AudioClip starting;
13        public AudioClip rolling;
14        public AudioClip stopping;
15
16        [Header("pitch parameter")]
17        public float flatoutSpeed = 20.0f;
18        [Range(0.0f, 3.0f)]
19        public float minPitch = 0.7f;
20        [Range(0.0f, 0.1f)]
21        public float pitchSpeed = 0.05f;
22
23        private AudioSource _source;
24        private WheelVehicle _vehicle;
25
26        void Start () {
27            _source = GetComponent<AudioSource>();
28            _vehicle = GetComponent<WheelVehicle>();
29        }
30
31        void Update () {
32            if (_vehicle.Handbrake && _source.clip == rolling)
33            {
34                _source.clip = stopping;
35                _source.Play();
36            }
37
38            if (!_vehicle.Handbrake && (_source.clip == stopping || _source.clip == null))
39            {

```

9

10

Class Suspension

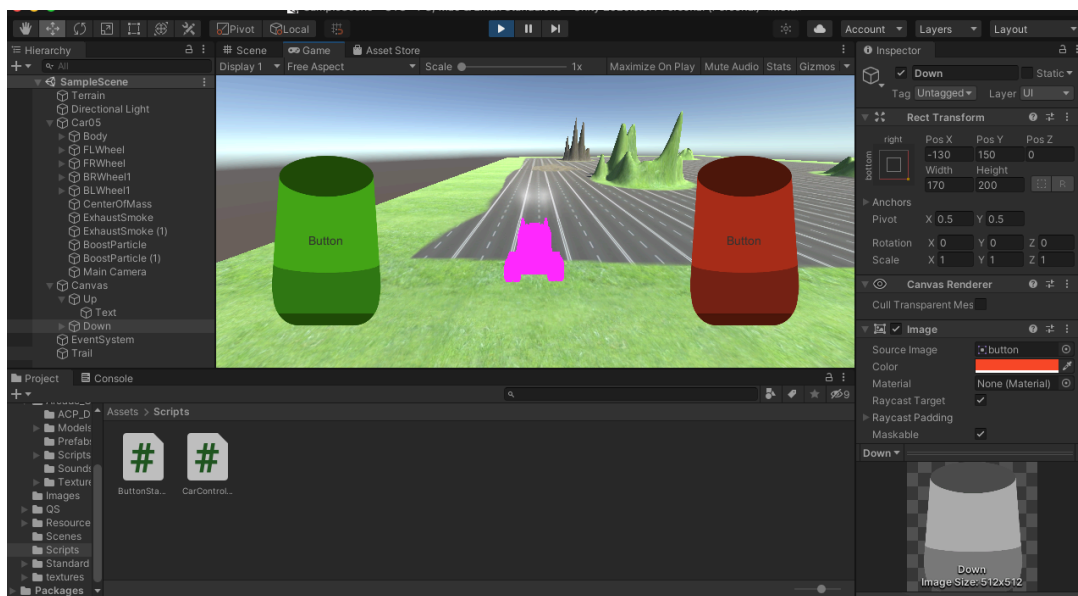
```

ButtonStateManager.cs  CarController.cs  CameraFollow.cs  EngineSoundManager.cs
Users > md1010 > Documents > Semester 6 > Komputasi Multimedia Project > UTS > UTS > Assets > Arcade_Car_Phys

8      public class Suspension : MonoBehaviour {
9
10     public GameObject _wheelModel;
11     private WheelCollider _wheelCollider;
12
13     [SerializeField] Vector3 localRotOffset;
14
15     private float lastUpdate;
16
17     void Start()
18     {
19         lastUpdate = Time.realtimeSinceStartup;
20
21         _wheelCollider = GetComponent<WheelCollider>();
22     }
23
24     void FixedUpdate()
25     {
26         if (Time.realtimeSinceStartup - lastUpdate < 1f/60f)
27         {
28             return;
29         }
30         lastUpdate = Time.realtimeSinceStartup;
31
32         if (_wheelModel && _wheelCollider)
33         {
34             Vector3 pos = new Vector3(0, 0, 0);
35             Quaternion quat = new Quaternion();
36             _wheelCollider.GetWorldPose(out pos, out quat);
37
38             _wheelModel.transform.rotation = quat;
39             _wheelModel.transform.localRotation *= Quaternion.Euler(localRotOffset);
40             _wheelModel.transform.position = pos;
41
42             WheelHit wheelHit;

```

Output



https://youtu.be/sgtZVxxUZ_k

<https://github.com/iqbal1922/Komputasi-Multimedia/tree/master>

-- Selamat Mengerjakan --