KOMPUTASI MULTIMEDIA

MODUL 15
CONTROLLING 3D ANIMATIONS

Oleh:

KINANTI PERMATA PUTRI NIM. 1841720022

TI-3D

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

JUNI 2021

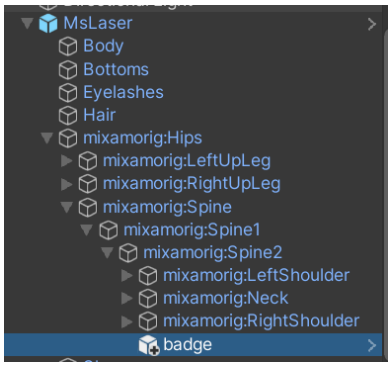
A. Praktikum

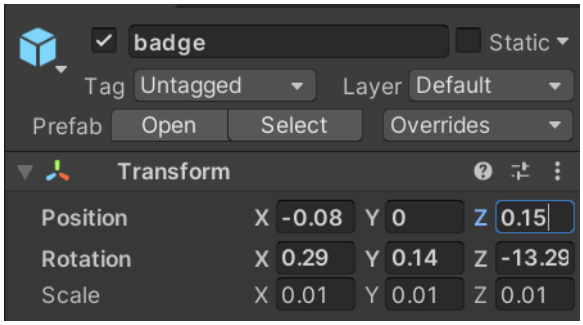
1. Menambahkan rigid props ke dalam karakter animasi

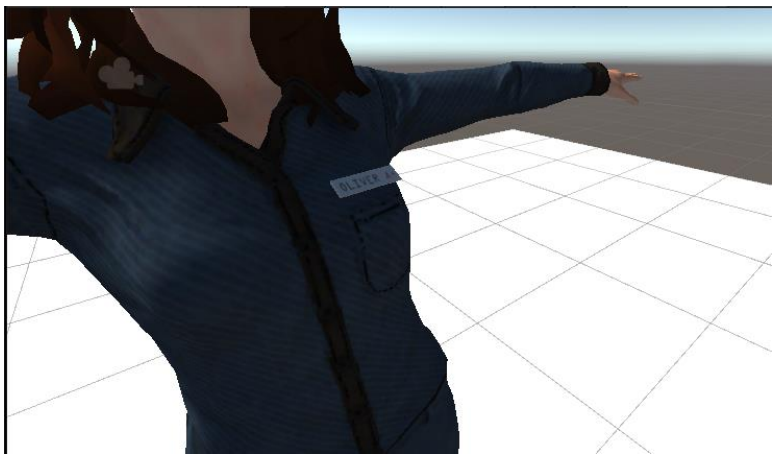
   1) Menggunakan paket asset dari folder 1362_07_06.

   2) Buatlah projek baru unity 3D. Import paket Props.unitypackage. Kemudian dari Project view, buka mecanimPlayground.

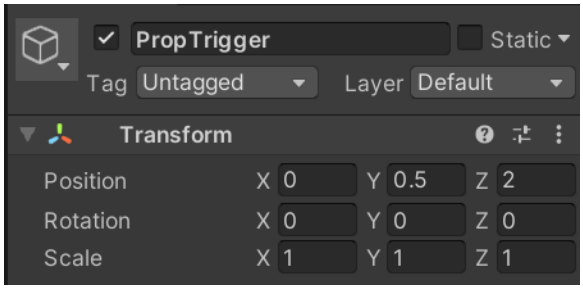
   3) Dari project view, masukkan badge prop kescene dengan cara drag kedalam hierarchy view. Kemudian, jadikan badge tersebut sebagai child dari mixamorig:Spine2. Kemudian, ubahlah position dan rotationnya seperti gambar berikut ini :



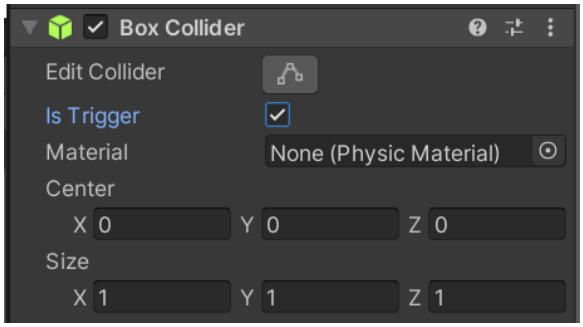




4) Catatlah nilai position dan rotation dari badge di dalam notepad, kemudian hapus objek badge dari hierarchy.
5) Tambahkan Cube baru (Create → 3D Object → Cube), ganti nama menjadi PropTrigger dan ubah position menjadi X = 0; Y = 0.5; Z = 2.



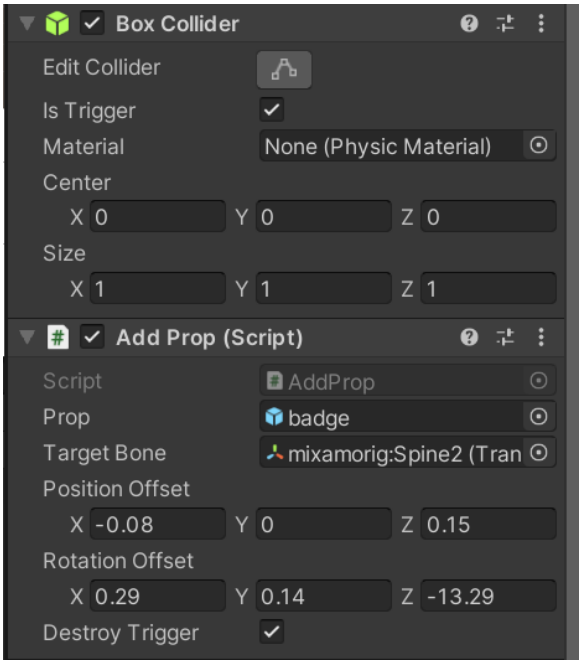
6) Pada Inspector dari PropTrigger, centangkan Is Trigger dari komponen Box Collider.



7) Buatlah C# Script dan masukkan code dibawak ini. Kemudian, ubahlah namanya menjadi AddProp.cs dan drag kedalam PropTrigger.

```
0 references
public class AddProp : MonoBehaviour
{
    3 references
    public GameObject prop;
    5 references
    public Transform targetBone;

    1 reference
    public Vector3 positionOffset;

    1 reference
    public Vector3 rotationOffset;

    1 reference
    public bool destroyTrigger = true;

    0 references
    void OnTriggerEnter (Collider collision) {
        if (targetBone.IsChildOf(collision.transform)) {
            bool checkProp = false;
            foreach(Transform child in targetBone) {
                if(child.name == prop.name)
                    checkProp = true;
            }
            if(!checkProp) {
                GameObject newProp;
                newProp = Instantiate(prop, targetBone.position, targetBone.rotation) as GameObject;
                newProp.name = prop.name;
                newProp.transform.parent = targetBone;
                newProp.transform.localPosition += positionOffset;
                newProp.transform.localEulerAngles += rotationOffset;
                if(destroyTrigger)
                    Destroy(gameObject);
            }
        }
    }
```

8) Pilih PropTrigger pada Hierarchy, kemudian ubahlah seperti gambar dibawah ini :



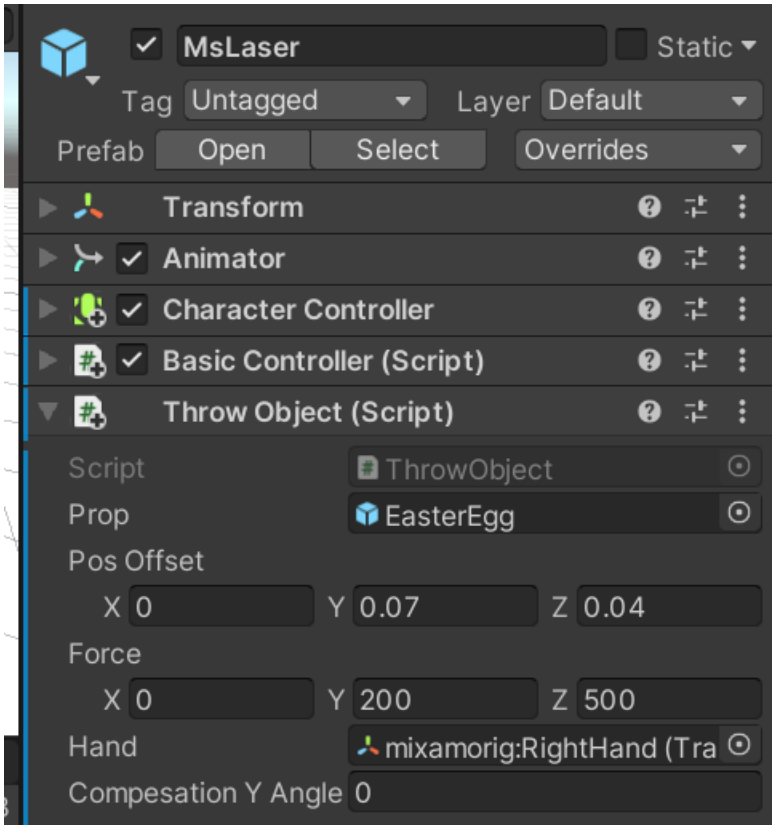
9) Play scene. Gunakan tombol 'WASD' untuk berjalan menuju PropTrigger. Jika PropTrigger ditabrak, maka badge (yang tadi dihapus) akan muncul dan PropTrigger akan hilang → cek Hierarchy.

2. Menggunakan Animation Event untuk melempar sebuah objek
    1) Menggunakan paket aseet unity dari folder 1362_07_07
    2) Buat project baru unity 3D.
    3) Import Throwing.unitypackage dari folder asset. Kemudian, buka mecanimPlayground.
    4) Play scene dan tekan tombol ‘F’ pada keyboard. Karakter akan bergerak ketika melempar sesuatu dengan tangan kanannya.
    5) Buatlah script C# baru dengan nama ThrowObject.cs. Kemudian, masukkan source code seperti dibawah ini :

```
public class ThrowObject : MonoBehaviour
{
    // Start is called before the first frame update
    1 reference
    public GameObject prop;
    14 references
    private GameObject proj;

    1 reference
    public Vector3 posOffset;

    1 reference
    public Vector3 force;

    3 references
    public Transform hand;

    1 reference
    public float compesationYAngle = 0f;

    0 references
    public void Prepare () {
        proj = Instantiate(prop, hand.position, hand.rotation) as GameObject;
        if(proj.GetComponent<Rigidbody>())
            Destroy(proj.GetComponent<Rigidbody>());
        proj.GetComponent<SphereCollider>().enabled = false;
        proj.name = "projectile";
        proj.transform.parent = hand;
        proj.transform.localPosition = posOffset;
        proj.transform.localEulerAngles = Vector3.zero;
    }

    0 references
    public void Throw() {
        Vector3 dir = transform.rotation.eulerAngles;
        dir.y += compesationYAngle;
        proj.transform.rotation = Quaternion.Euler(dir);
        proj.transform.parent = null;
        proj.GetComponent<SphereCollider>().enabled = true;
        Rigidbody rig = proj.AddComponent<Rigidbody>();
        Collider projCollider = proj.GetComponent<Collider>();
        Collider col = proj.GetComponent<Collider>();
        Physics.IgnoreCollision(projCollider, col);
        rig.AddRelativeForce(force);
    }
}
```

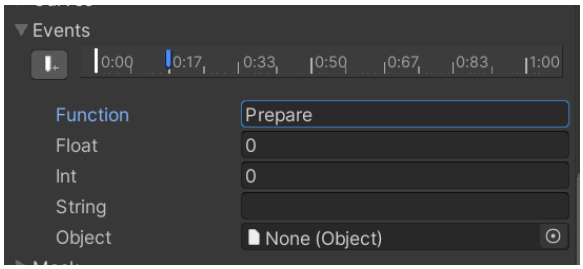
6) Drag ThrowObject.cs kedalam MsLaser pada Hierarchy.

7) Buka Inspector dari MsLaser. Kemudian, ubahlah seperti berikut ini :

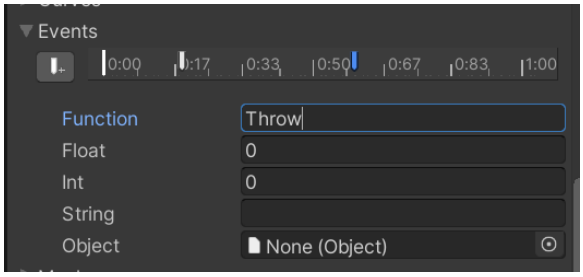


8) Pada project view, pilih file Swat@toss_grenade. Kemudian, pada Inspectornya, pilih bagian Animation dan expand bagian Event.

9) Klik tombol '+' untuk menambahkan animasi. Ubahlah playhead menuju detik 0:17. Kemudian, ubahlah Function menjadi Prepare dan klik tombol Apply.

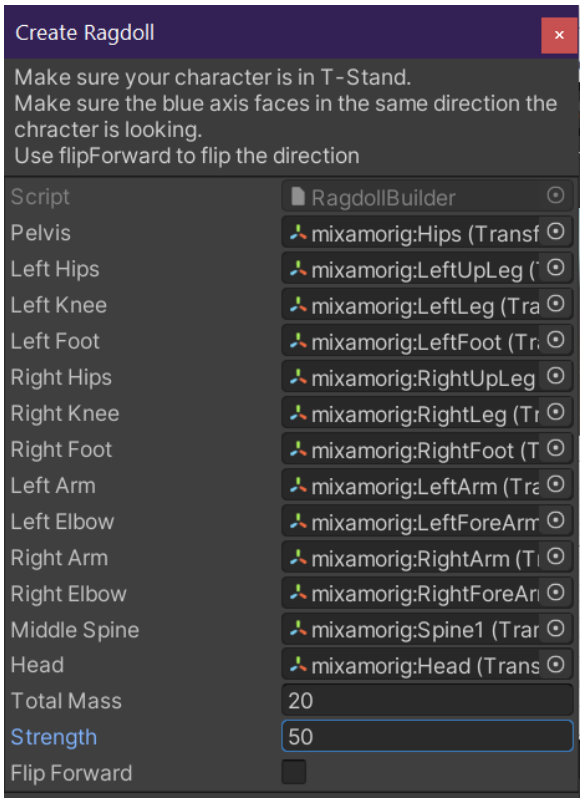


10) Klik tombol '+' untuk menambahkan animasi. Ubahlah playhead menuju detik 1:00. Kemudian, ubahlah Function menjadi Throw dan klik tombol Apply.



11) Play scene. Karakter akan melempar Easter Egg ketika tombol 'F' ditekan. Karakter akan bersiap untuk melempar pada detik ke 0.17 (Prepare) dan mulai melempar pada menit ke 1.00 (Throw).

3. Membuat Ragdoll untuk sebuah karakter
   1) Menggunakan paket asset dari folder 1362_07_08.
   2) Buatlah project baru unity 3D.
   3) Import Ragdoll.unitypackage. Kemudian, buka mecanimPlayground.
   4) Buatlah objek Ragdoll (GameObject → 3D Object → Ragdoll…). Kemudian, pilih transform seperti berikut ini :



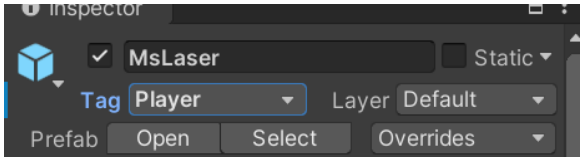
5) Buatlah script C# baru dengan nama RagdollCharacter.cs. Kemudian, isikan source code seperti dibawah ini dan drag kedalam MsLaser.

```
using UnityEngine;
using System.Collections;
/* ----------------------------------------
 * class to demonstrate how to apply Ragdoll physics
 * to a character previously set up with Ragdoll Wizard
 */
0 references
public class RagdollCharacter : MonoBehaviour {
    /* ----------------------------------------
     * At Start, deactivate all components that allow for ragdoll physics.
     * Also starting a coroutine that restores the character after 5 seconds
     */
    0 references
    void Start () {
        // Call DeactivateRagdoll function
        DeactivateRagdoll();
    }

    /* ----------------------------------------
     * A function to activate all components that allow for ragdoll physics
     */
    0 references
    public void ActivateRagdoll(){
        // Disable Character Controller component
        gameObject.GetComponent<CharacterController> ().enabled = false;

        //  Disable character's Basic Controller component (a C# script that controls Mecanim syst
        gameObject.GetComponent<BasicController> ().enabled = false;

        //  Disable Animator component
        gameObject.GetComponent<Animator> ().enabled = false;

        // Find every Rigidbody in character's hierarchy
        foreach (Rigidbody bone in GetComponentsInChildren<Rigidbody>()) {
            // Set bone's rigidbody component as not kinematic
            bone.isKinematic = false;

            //Enable collision detection for rigidbody component
            bone.detectCollisions = true;
        }

        // Find every Collider in character's hierarchy
        foreach (Collider col in GetComponentsInChildren<Collider>()) {
            // Enable Collider
            col.enabled = true;
        }
```

```
        // Start coroutine to restore character
        StartCoroutine (Restore ());

    }

    /* ---------------------------------------
     * A function to deactivate all components that allow for ragdoll physics
     */
    2 references
    public void DeactivateRagdoll(){
        // Enable Character Controller component
        gameObject.GetComponent<BasicController>().enabled = true;

        //  Enable Animator component
        gameObject.GetComponent<Animator>().enabled = true;

        // Position character at Spawnpoint gameobject's position
        transform.position = GameObject.Find("Spawnpoint").transform.position;

        // Rotate character according to Spawnpoint gameobject's rotation
        transform.rotation = GameObject.Find("Spawnpoint").transform.rotation;

        // Find every Rigidbody in character's hierarchy
        foreach(Rigidbody bone in GetComponentsInChildren<Rigidbody>()){
            // Set bone's rigidbody component as  kinematic
            bone.isKinematic = true;
            // Disable collision detection for rigidbody component
            bone.detectCollisions = false;
    }

        // Find every Collider in character's hierarchy
        foreach(Collider col in GetComponentsInChildren<Collider>()){
            // Disable Collider
            col.enabled = false;
        }

        //  Enable character's Basic Controller component (a C# script that controls Mecanim syste
        gameObject.GetComponent<CharacterController>().enabled = true;

    }
```

```
    1 reference
    IEnumerator Restore(){
        // Wait for five seconds
        yield return new WaitForSeconds(5);

        // Deactivate Ragdoll
        DeactivateRagdoll();
    }
}
```

6) Pilih MsLaser. Kemudian, pada Inspector view, ubahlah tag menjadi Player.



7) Buatlah script C# baru dengan source code seperti dibawah ini. Ubahlah namanya menjadi Landmine.cs. Kemudian, drag kedalam Landmine pada Hierarchy.

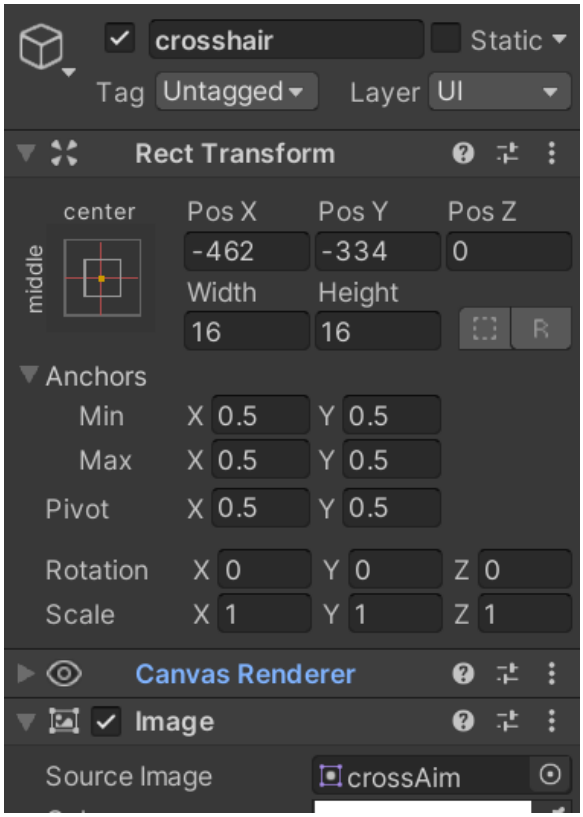
```
using UnityEngine;
using System.Collections;
/* ----------------------------------------
 * class to demonstrate how to trigger and apply and Explosion force to
 * a character featuring Ragdoll physics
 */
0 references
public class Landmine : MonoBehaviour {
    // Float variable for the explosion's radius
    2 references
    public float range = 2f;

    // Float variable for the explosion's force
    1 reference
    public float force = 2f;

    // Float variable for the explosion's Upwards modifier
    1 reference
    public float up = 4f;

    // Private bool for enabling/disabling the effects of a collision with the trigger
    3 references
    private bool active = true;

    /* ----------------------------------------
     * If a game object with the'Player' tag enters the Trigger collider, and 'active' is true,
     * activate character's ragdoll physics, apply explosion force to it, deactivate trigger and
     * coroutine to re-activate it.
     */
    0 references
    void  OnTriggerEnter ( Collider collision  ){
        if(collision.gameObject.tag == "Player" && active){
            //IF gameobject that has collided with trigger has 'Player' tag and 'active' is true
            active = false;

            // Start coroutine to reactivate trigger
            StartCoroutine(Reactivate());

            // Activate ragdoll physics on character through its RagdollCharacter component
            collision.gameObject.GetComponent<RagdollCharacter>().ActivateRagdoll();

            // Create Vector for the explosion's position
            Vector3 explosionPos = transform.position;

            // Get all colliders within explosion radius
            Collider[] colliders = Physics.OverlapSphere(explosionPos, range);

            // For each collider within explosion radius...
            foreach (Collider hit in colliders) {
                if (hit.GetComponent<Rigidbody>())
                    // IF collider object features a rigidbody component, add explosion force to it
                    hit.GetComponent<Rigidbody>().AddExplosionForce(force, explosionPos, range, up);
            }
        }
    }

    /* ----------------------------------------
     * A function to reactivate the trigger after two seconds
     */
    1 reference
    IEnumerator Reactivate(){
        // Wait for two seconds
        yield return new WaitForSeconds(2);

        // set 'active' as true
        active = true;
    }
}
```

8) Play scene. Gunakan tombol ‘WASD’ pada keyboard untuk menjalankan karakter. Jika karakter menabrak Landmine, maka akan terjadi sebuah ledakan sehingga karakter akan terlempar menjauh. Selain itu, menyebabkan pergerakan badan akan mirip seperti ragdoll saat terjatuh.

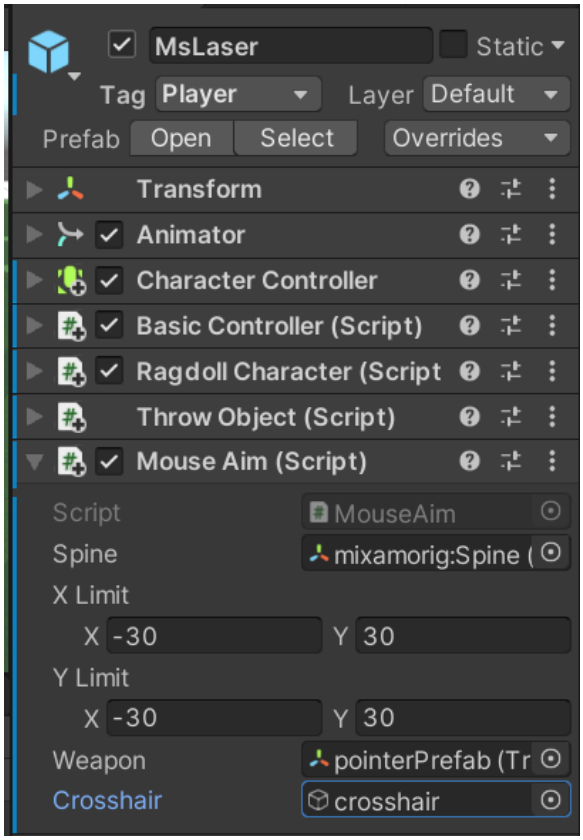
4. Membuat putaran torso pada karakter untuk mengarahkan senjata
   1) Menggunakan paket asset dari folder 1362_07_09.
   2) Buatlah project baru unity 3D.
   3) Import AimPointer.unitypackage. Kemudian, buka mecanimPlayground.
   4) Buat script C# baru dengan nama MouseAim.cs dengan source code seperti dibawah ini. Kemudian, drag kedalam MsLaser.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MouseAim : MonoBehaviour
{
    public Transform spine;
    private float xAxis = 0f;
    private float yAxis = 0f;
    public Vector2 xLimit = new Vector2(-30f,30f);
    public Vector2 yLimit = new Vector2(-30f,30f);

    public Transform weapon;
    public GameObject crosshair;
    private Vector2 aimLoc;

    public void LateUpdate(){
        yAxis += Input.GetAxis("Mouse X");
        yAxis = Mathf.Clamp(yAxis, yLimit.x, yLimit.y);
        xAxis -= Input.GetAxis("Mouse Y");
        xAxis = Mathf.Clamp(xAxis, xLimit.x, xLimit.y);
        Vector3 corr = new Vector3(xAxis, yAxis, spine.localEulerAngles.z);
        spine.localEulerAngles = corr;
        RaycastHit hit;
        Vector3 fwd = weapon.TransformDirection(Vector3.forward);
        if(Physics.Raycast (weapon.position, fwd, out hit)){
            print(hit.transform.gameObject.name);
            aimLoc = Camera.main.WorldToScreenPoint(hit.point);
            crosshair.SetActive(true);
            crosshair.transform.position = aimLoc;
        } else {
            crosshair.SetActive(false);
        }
        Debug.DrawRay(weapon.position, fwd, Color.red);
    }
}
```

5) Buatlah objek Image baru (Create → UI → Image). Ubah namanya menjadi crosshair. Pada Inspector, ubah Width dan Height menjadi 16. Kemudian ubah Source Image menjadi crossAim.



6) Buka Inspector dari MsLaser, kemudian ubahlah beberapa komponen seperti berikut ini :



7) Play scene. Torso dari karakter dapat digerakkan dengan cara menggerakkan mouse. Selain itu, crosshair GUI texture ditampilkan diatas objek yang bertujuan sebagai pointer.

B. TUGAS PRAKTIKUM

Implementasikan 4 projek dari modul ini. Kemudian, buatlah laporan secara detail (termasuk penjelasan source code) dengan bahasa yang mudah dimengerti.

Link Youtube: https://youtu.be/1vJhwGoKuiU

Link Github: https://github.com/kinanpermata/TI-3D_17_Komputasi-Multimedia