

TEST ASSESMENT FULL STACK WEB

KELOMPOK 2

Tugas :

- Buatlah aplikasi perhitungan permainan tennis
- Page untuk pertandingan match beserta button penambahan nilai dan posisi bola.

Stack : Aplikasi menggunakan Bahasa yang biasa digunakan.

UI/ UX : Untuk tampilan bisa di touch up sesuai dengan kebutuhan. Makin menarik lebih baik. Figma dibawah hanya untuk gambaran saja.

Cakupan test : Front End dan Backend.

Bisnis dari permainan tennis :

Detail : <https://id.wikihow.com/Menghitung-Skor-Permainan-Tenis>

Sebuah game akan dimenangkan jika salah satu pemain (atau tim, jika Anda bermain ganda) mendapatkan 4 poin. Sebuah poin didapatkan sampai ada pemain yang memukul bola ke net atau keluar dari lapangan pada saat seorang pemain melakukan servis yang dibalas oleh pemain lawan dan seterusnya. Ingatlah bahwa sebuah game biasanya membutuhkan 7 poin atau bahkan lebih untuk selesai, jika ada saat dimana satu pemain mendapatkan 3 poin dan pemain yang lain mendapatkan 4 poin. Setiap poin yang didapatkan akan di ubah menjadi angka berikut untuk setiap pemain^[1] :

- poin pertama ditulis “15 poin”
- poin kedua ditulis “30 poin”
- poin ketiga ditulis “40 poin”
- poin ke empat ditulis GAME (menandakan bahwa permainan telah berakhir)

..... Detail di wiki.

Poin penilaian adalah

1. Bisa menjelaskan secara bagaimana program dijalankan , dan perhitungannya.
2. Data hasil match di simpan di table dan di tunjukan dalam table. Termasuk nama yang bertanding.
3. Perhatikan jika ada game “deuce”.
- 4.

GAME TENNIS SEDERHANA MENGGUNAKAN BAHSA C#

- **UI/UX**



Dalam UI ini ada beberapa komponen yaitu sebagai berikut :

1. Net
2. Raket
3. Player
4. Bola
5. Target
6. Lampu
7. Lapangan
8. Dinding

- **Penjelasan kode program**

- **Player.cs**

```
0 references
public class Player : MonoBehaviour
{
    4 references
    public Transform aimTarget; // target di mana kita bertujuan untuk mendaratkan bola yang di pukul ol
    2 references
    float speed = 3f; // kecepatan Bergeraknya bola
    0 references
    float force = 13; // kekuatan memukul bola

    6 references
    bool hitting; // untuk mengetahui apakah kita memukul bola atw tidak

    1 reference
    public Transform ball; // bola
    3 references
    Animator animator;

    2 references
    Vector3 aimTargetInitialPosition; // posisi awal bola

    4 references
    ShotManager shotManager;
    5 references
    Shot currentShot; // bidikan ketika bola di pukul
```

Pada codingan di atas, membuat sebuah perintah untuk player agar dapat bergerak, memukul bola dan bagaimana bola itu dapat bergerak ke target yang telah ditentukan

```

0 references
private void Start()
{
    animator = GetComponent<Animator>();
    aimTargetInitialPosition = aimTarget.position;
    shotManager = GetComponent<ShotManager>();
    currentShot = shotManager.topSpin;
}

```

Lalu membuat private void start, dimana ketika awal mula bola akan di pukul oleh player

```

0 references
void Update()
{
    float h = Input.GetAxisRaw("Horizontal");
    float v = Input.GetAxisRaw("Vertical");

    if (Input.GetKeyDown(KeyCode.F))
    {
        hitting = true;
        currentShot = shotManager.topSpin;
    }
    else if (Input.GetKeyUp(KeyCode.F))
    {
        hitting = false;
    }

    if (Input.GetKeyDown(KeyCode.E))
    {
        hitting = true;
        currentShot = shotManager.flat;
    }
    else if (Input.GetKeyUp(KeyCode.E))
    {
        hitting = false;
    }

    if (hitting)
    {
        aimTarget.Translate(new Vector3(h, 0, 0) * speed * 2 * Time.deltaTime);
    }

    if ((h != 0 || v != 0) && !hitting)
    {
        transform.Translate(new Vector3(h, 0, v) * speed * Time.deltaTime);
    }
}

```

Lalu membuat sebuah perintah keyboard untuk mengarahkan bola Ketika di pukul oleh player dan dapat menggerakan target bidikan kita sendiri, untuk memudahkan player mendapatkan point.

```
0 references
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Ball"))
    {
        Vector3 dir = aimTarget.position - transform.position;
        other.GetComponent<Rigidbody>().velocity = dir.normalized * currentShot.hitForce + new Vector3(0, currentShot.upForce, 0);

        Vector3 ballDir = ball.position - transform.position;
        if (ballDir.x >= 0)
        {
            animator.Play("forehand");
        }
        else
        {
            animator.Play("backhand");
        }

        aimTarget.position = aimTargetInitialPosition;
    }
}
```

Kemudian yang terakhir membuat perintah bagaimana agar bola itu dapat bergerak kearah target, jadi ketika bola bertabrakan ke raket player maka bola tersebut akan bergerak menuju kearah lawan.

- **Ball.cs**

```

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

0 references
public class Ball : MonoBehaviour
{
    2 references
    Vector3 initialPos;

    0 references
    private void Start()
    {
        initialPos = transform.position;
    }

    0 references
    private void OnCollisionEnter(Collision collision)
    {
        if (collision.transform.CompareTag("Wall"))
        {
            GetComponent<Rigidbody>().velocity = Vector3.zero;
            transform.position = initialPos;
        }
    }
}

```

Pada codingan di atas, membuat sebuah perintah kepada bola, dimana jika bola tersebut menabrak dinding atau out maka bola tersebut akan kembali ke titik 0 dimana awal mula bola itu di letakan.

- **Bot.cs**

```

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

0 references
public class Bot : MonoBehaviour
{
    1 reference
    float speed = 40;
    3 references
    Animator animator;
    2 references
    public Transform ball;
    0 references
    public Transform aimTarget;

    2 references
    public Transform[] targets;

    0 references
    float force = 13;
    3 references
    Vector3 targetPosition;

    3 references
    ShotManager shotManager;
}

```

Pada codingan di atas, membuat sebuah perintah mengatur kecepatan bergeraknya bot dan kecepatan benturan bola yang dipukul oleh bot

```

0 references
void Start()
{
    targetPosition = transform.position;
    animator = GetComponent<Animator>();
    shotManager = GetComponent<ShotManager>();
}

```

Lalu membuat perintah dimana ketika mulai game nya posisi pemain akan pada posisi awal, membuat sebuah animasi bagaimana pemain itu bergerak dan mereferensi ke ShotManager untuk menembak bola sesuai yang kita arahkan atau inginkan.

```

0 references
void Update()
{
    Move();
}

```

```

1 reference
void Move()
{
    targetPosition.x = ball.position.x;
    transform.position = Vector3.MoveTowards(transform.position, targetPosition, speed * Time.deltaTime);
}

```

Lalu membuat perintah update dimana ketika update posisi bot akan kembali ke sumbu X sehingga bot hanya bergerak pada sumbu X.

```

1 reference
Vector3 PickTarget()
{
    int randomValue = Random.Range(0, targets.Length);
    return targets[randomValue].position;
}

```

Lalu membuat sebuah target array yang dimulai dari nilai 0 sampai dengan -1 dan mengembalikan target yang dipilih

```

1 reference
Shot PickShot()
{
    int randomValue = Random.Range(0, 2);
    if (randomValue == 0)
    {
        return shotManager.topSpin;
    }
    else
    {
        return shotManager.flat;
    }
}

```

Lalu membuat perintah bagaimana bola tersebut akan berpindah ketika bola tersebut di pukul oleh raket bot maka bola tersebut akan melambung ke atas dan mengarah ke pemain lawan, jika nilai nya sama dengan 0 maka jenis tembakan nya melambung ke atas,

```

0 references
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Ball"))
    {
        Shot currentShot = PickShot();

        Vector3 dir = PickTarget() - transform.position;
        other.GetComponent<Rigidbody>().velocity = dir.normalized * currentShot.hitForce + new Vector3(0, currentShot.upForce, 0);
        Vector3 ballDir = ball.position - transform.position;
        if (ballDir.x >= 0)
        {
            animator.Play("forehand");
        }
        else
        {
            animator.Play("backhand");
        }
    }
}

```

Lalu yang terakhir, membuat animasi pukulan ke kanan dan ke kiri, ketika bola ada di sebelah kanan bot maka raket tersebut akan memukul dari sebelah kanan dan jika bola

ada disebalh kiri bot maka raket tersebut akan memukul dari sebelah kiri, lalu dapat membidik kemana bola tersebut akan di arahkan

- **Score.cs**
Masih belum berhasil