

UAS – GAME PROJECT

PARA PENCARI RTX

Anggota Kelompok :

Daffa Hasanah Arkaan (L0123043)

Farhan Cahya Permana (L0123051)

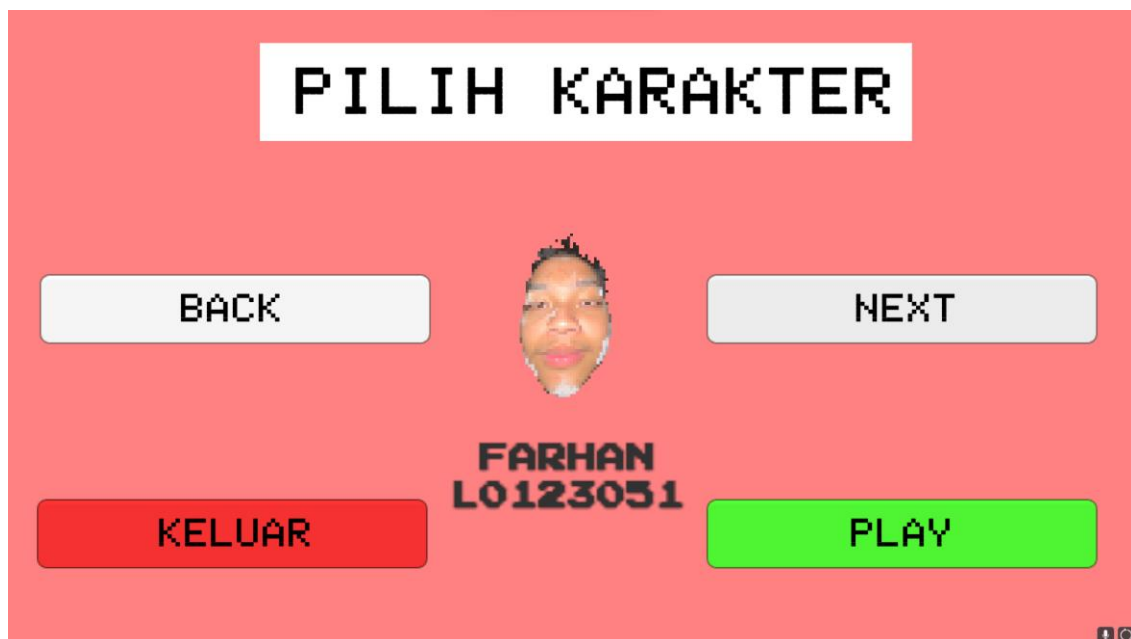
Fawwaz Naufal Maulana (L0123055)

Deskripsi

Para Pencari RTX adalah suatu game platform yang dibuat menggunakan Unity yang diintegrasikan dengan kode atau pemrograman C#. Player atau tokoh utama dari game ini adalah kami bertiga (Daffa, Farhan, dan Opal). Game ini memiliki 2 level yang di setiap levelnya terdapat rintangan dan player dapat mengumpulkan item berupa RTX yang sudah disediakan di Map.

How to play + Penjelasan Source Code

Menu Pilih Karakter



Pertama pilih karakter yang ingin dimainkan, Klik Next untuk karakter selanjutnya dan Back untuk karakter sebelumnya. Klik Play untuk memulai permainan, klik keluar untuk keluar dari game.

Source Code Menu Pilih Karakter (KarakterManager.cs)

```
0 references
public class KarakterManager : MonoBehaviour
{
    3 references
    public DataKarakter karakterDB;

    1 reference
    public Text namaKarakter;
    1 reference
    public SpriteRenderer gambarKarakter;
    12 references
    private int pilihan = 0;
```

Deklarasi variabel karakterDB (database karakter yang berisi text (nama karakter) dan gambar (jpg karakter)), namaKarakter, gambarKarakter, dan pilihan.

```
void Start()
{
    if(!PlayerPrefs.HasKey("pilihan"))
    {
        pilihan = 0;
    }

    else
    {
        Load();
    }
    UpdateKarakter(pilihan);
}
```

void Start() adalah fungsi yang berjalan ketika program (dalam konteks ini Scene) dimulai, kode diatas berisi if else / conditional untuk mengecek apakah sebelumnya User sudah pernah memilih karakter atau belum, jika sudah maka akan di Load dengan fungsi Load() lalu karakter akan di update dengan fungsi UpdateKarakter dengan parameter pilihan.

```
0 references
public void NextOption()
{
    pilihan++;

    if(pilihan >= karakterDB.JumlahKarakter)
    {
        pilihan = 0;
    }
    UpdateKarakter(pilihan);
    Save();
}

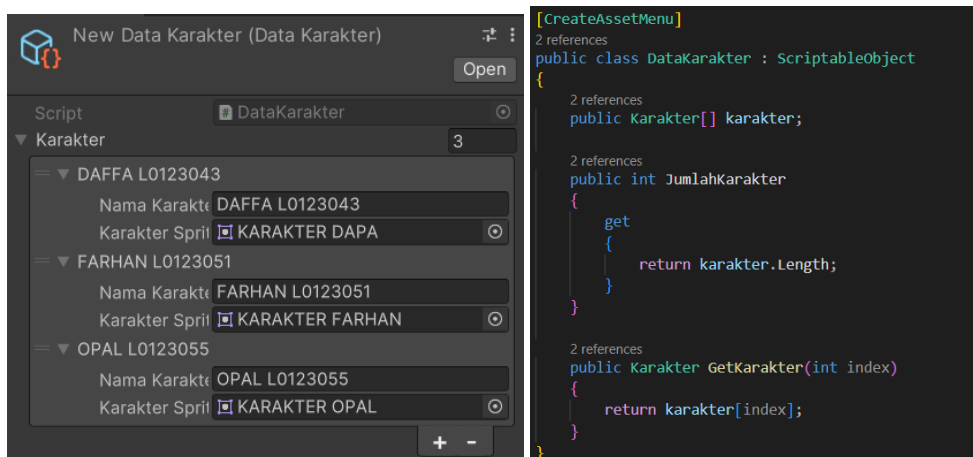
0 references
public void BackOption()
{
    pilihan--;

    if(pilihan < 0)
    {
        pilihan = karakterDB.JumlahKarakter - 1;
    }
    UpdateKarakter(pilihan);
    Save();
}

3 references
private void UpdateKarakter (int pilihan)
{
    Karakter karakter = karakterDB.GetKarakter(pilihan);
    gambarKarakter.sprite = karakter.karakterSprite;
    namaKarakter.text = karakter.namaKarakter;
}
```

Fungsi NextOption() untuk menambah int pilihan dan mengupdate karakter lalu disave, BackOption() untuk mengurangi int pilihan dan mengupdate karakter lalu disave, dan UpdateKarakter dengan parameter (int pilihan) untuk mengubah karakterDB. Urutan karakter yang dipengaruhi oleh int pilihan di setting pada Unity

KarakterDB (Database) yang ada didalam Unity



Database karakter tersebut dibuat dengan script DataKarakter.cs



Fungsi Load() untuk me-Load atau Get KarakterDB sesuai dengan pilihan, sedangkan fungsi Save untuk men-Save atau Set. Fungsi ChangeScene untuk me-Load scene yang ada, urutan scene ini disetting didalam Unity.

In Game

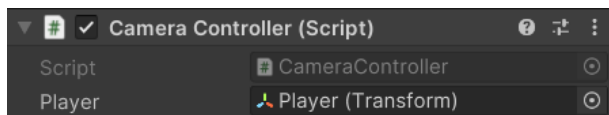


Pada game, player dapat menggerakkan karakter dengan WASD atau Arah Panah (Arrow Key) dan Spasi pada keyboard. Tugas Player adalah untuk mendapatkan RTX yang sudah ada di maps, selain itu ada obstacle-obstacle atau trap yang harus dilewati atau dihindari player, jika player terkena trap maka player akan mengulang dari awal. Pada game terdapat tombol “KELUAR” untuk langsung keluar dari game jika Player sudah capek melewati trap yang ada.

Source Code Game

```
5 public class CameraController : MonoBehaviour
6 {
7     2 references
8     [SerializeField] private Transform player;
9
10    0 references
11    private void Update()
12    {
13        transform.position = new Vector3(player.position.x, player.position.y, transform.position.z);
14    }
15 }
```

Diatas merupakan source code dari **CameraController.cs**, code ini berfungsi untuk mengubah posisi Main Camera untuk mengikuti sebuah objek, dalam hal ini objeknya adalah “Player”, menggunakan [SerializeField] agar bisa memilih GameObject yang akan menjadi “Player”.



```

8      1 reference
      public DataKarakter karakterDB;
9
10     0 references
      public Text namaKarakter;
11     1 reference
      public SpriteRenderer gambarKarakter;
12     3 references
      private int pilihan = 0;
13     // Start is called before the first frame update
14     0 references
      void Start()
15     {
16         if(!PlayerPrefs.HasKey("pilihan"))
17         {
18             pilihan = 0;
19         }
20
21         else
22         {
23             Load();
24         }
25         UpdateKarakter(pilihan);
26     }
27
28     1 reference
      private void UpdateKarakter (int pilihan)
29     {
30         Karakter karakter = karakterDB.GetKarakter(pilihan);
31         gambarKarakter.sprite = karakter.karakterSprite;
32     }
33
34     1 reference
      private void Load()
35     {
36         pilihan = PlayerPrefs.GetInt("pilihan");
37     }
38 }

```

Kode diatas merupakan kode dari **Player.cs** dimana kode tersebut berfungsi untuk me-Load Assets Player yang sudah dipilih / Karakter yang sudah dipilih pada menu awal tadi.

```

7      1 reference
      [SerializeField] AudioSource suaraLoncat;
8      1 reference
      [SerializeField] float kecepatan = 14f;
9      5 references
      private Rigidbody2D rb;
10     3 references
      private BoxCollider2D coll;
11     1 reference
      [SerializeField] private LayerMask jumpableGround;
12     // Start is called before the first frame update
13     0 references
      private void Start()
14     {
15         rb = GetComponent<Rigidbody2D>();
16         coll = GetComponent<BoxCollider2D>();
17     }
18     // Update is called once per frame
19     0 references
      private void Update()
20     {
21         float dirX = Input.GetAxisRaw("Horizontal");
22         rb.velocity = new Vector2(dirX * 7f, rb.velocity.y);
23
24         if ((Input.GetButtonDown("Jump") || Input.GetKeyDown("w") || Input.GetKeyDown("up")) && IsGrounded())
25         {
26             rb.velocity = new Vector2(rb.velocity.x, kecepatan);
27             suaraLoncat.Play();
28         }
29     }
30
31     1 reference
      private bool IsGrounded()
32     {
33         return Physics2D.BoxCast(coll.bounds.center, coll.bounds.size, 0f, Vector2.down, .1f, jumpableGround);
34     }
35 }

```

Kode diatas merupakan kode dari **PlayerMovement.cs** yang berfungsi untuk memberikan kendali kepada Player, pada kode terdapat Input.GetAxisRaw("Horizontal"); ini memanfaatkan Unity sebagai Game Engine, karena keyboard "a", "d" dan arrow kanan kiri sudah termasuk pada "Horizontal" di Unity. Sedangkan untuk gerakan Vertical atau Loncat, kami ketik manual dengan GetButtonDown("Jump") untuk spasi, dan GetKeyDown("w") dan ("up") dan kami AND (&&) kan dengan variabel boolean IsGrounded, dimana variabel tersebut mengecek apakah Player menyentuh tanah atau tidak, dengan LayerMask jumpableGround yang di setting di dalam Unity sebagai parameternya. Lalu ada AudioSource dengan [SerializeField] untuk memasukkan suara ketika Loncat

```

public class ItemCollector : MonoBehaviour
{
    1 reference
    [SerializeField] AudioSource suaraDapat;
    2 references
    public int rtx;
    1 reference
    [SerializeField] private Text rtxText;

    // Use OnTriggerEnter2D to handle trigger collisions
    0 references
    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.CompareTag("rtx"))
        {
            Destroy(other.gameObject);
            rtx++;
            rtxText.text = "RTX Collected : " + rtx;
            suaraDapat.Play();
        }
    }
}

```

Kode diatas merupakan kode dari **ItemCollector.cs** yang berfungsi untuk kendali agar item (RTX) dapat terambil dan skor pemain dapat bertambah. Pada kode terdapat `Destroy(other.gameObject)` untuk menghilangkan objek (RTX) ketika tersentuh oleh Player, lalu ada `int rtx` untuk skor player, dan akan bertambah jika Player menyentuh objek (RTX), lalu ada `AudioSource suaraDapat` untuk memberi sound effect ketika Player menyentuh objek (RTX).

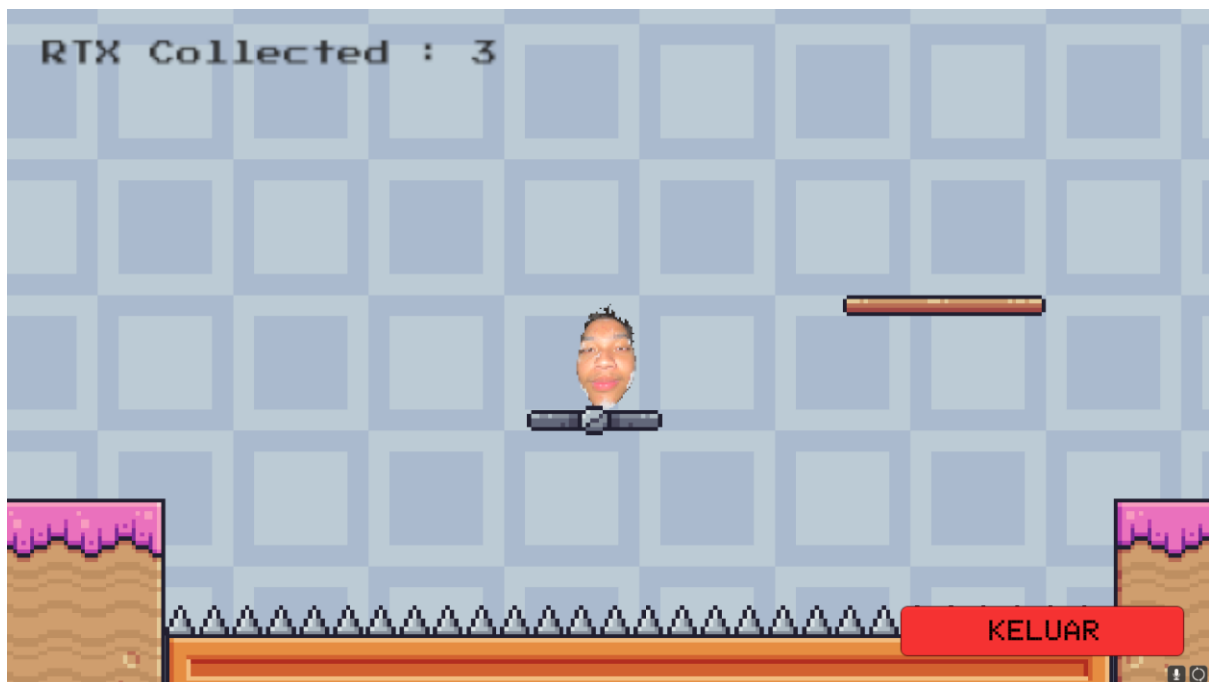
```

6 public class PlayerLife : MonoBehaviour
7 {
    1 reference
    [SerializeField] AudioSource suaraMati;
    2 references
    private Rigidbody2D rb;
    2 references
    private Animator anim;
    11 // Start is called before the first frame update
    0 references
    private void Start()
    {
        14 rb = GetComponent<Rigidbody2D>();
        15 anim = GetComponent<Animator>();
    }
    0 references
    private void OnCollisionEnter2D(Collision2D other)
    {
        19 if (other.gameObject.CompareTag("Trap"))
        {
            21 Mati();
            22 suaraMati.Play();
        }
    }
    1 reference
    private void Mati()
    {
        27 rb.bodyType = RigidbodyType2D.Static;
        28 anim.SetTrigger("death");
    }
    0 references
    private void RestartLevel()
    {
        33 SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }
}
35

```

Kode diatas merupakan kode dari **PlayerLife.cs** yang berfungsi apabila Player menyentuh Trap, kode ini membuat Player menjadi "static" atau tidak bisa digerakkan, dan menjalankan animasi "death" (menyusut) dan juga Game akan di Restart atau level tersebut akan direstart.

In Game diatas Moving Platform



Pada game diatas player bisa menempel pada platform dan platform bisa bergerak dari satu titik, ke titik yang lain (waypoints). Hal ini di kendalikan menggunakan source code **StickyPlatform.cs** dan **WaypointFollower.cs** dibawah ini.

```
5 public class StickyPlatform : MonoBehaviour
6 {
7     0 references
8     private void OnTriggerEnter2D(Collider2D other)
9     {
10         if(other.gameObject.name == "Player")
11         {
12             other.gameObject.transform.SetParent(transform);
13         }
14     }
15     0 references
16     private void OnTriggerExit2D(Collider2D other)
17     {
18         if(other.gameObject.name == "Player")
19         {
20             other.gameObject.transform.SetParent(null);
21         }
22     }
23 }
```

Pada kode **StickyPlatform.cs** diatas, gameObject bernama Player akan mengikuti transform (perpindahan) suatu objek apabila menyentuh objek tersebut dalam hal ini platform yang bergerak. Dan apabila sudah tidak menyentuh, maka Player tidak mengikuti objek apapun.

```

5 public class WaypointFollower : MonoBehaviour
6 {
7     3 references
8     [SerializeField] private GameObject[] waypoints;
9     5 references
10    private int currentWaypointIndex = 0;
11    1 reference
12    [SerializeField] private float speed = 2f;
13    0 references
14    private void Update()
15    {
16        if(Vector2.Distance(waypoints[currentWaypointIndex].transform.position, transform.position) < .1f)
17        {
18            currentWaypointIndex++;
19            if(currentWaypointIndex >= waypoints.Length)
20            {
21                currentWaypointIndex = 0;
22            }
23        }
24        transform.position = Vector2.MoveTowards(transform.position, waypoints[currentWaypointIndex].transform.position, Time.deltaTime * speed);
25    }
26 }

```

Pada kode **WaypointFollower.cs** di atas, objek di setting, untuk berpindah dari waypoint satu ke waypoint lainnya, disini memanfaatkan [SerializeField] agar bisa menginput waypoint di dalam Unity.

```

5 public class muter : MonoBehaviour
6 {
7     1 reference
8     [SerializeField] float xAngle = 0f;
9     1 reference
10    [SerializeField] float yAngle = 0f;
11    1 reference
12    [SerializeField] float zAngle = 1f;
13
14    // Update is called once per frame
15    0 references
16    void Update()
17    {
18        transform.Rotate(xAngle * Time.deltaTime, yAngle * Time.deltaTime, zAngle * Time.deltaTime);
19    }
20 }

```

Kode diatas adalah **muter.cs** yang berfungsi untuk mengatur suatu objek agar bisa berputar, dan memanfaatkan [SerializeField] agar bisa mengatur di dalam Unity kecepatan dan vektor arahnya.


```

6 public class Finish : MonoBehaviour
7 {
8     2 references
    private AudioSource suaraFinish;
9     2 references
    private bool selesaiLevel = false;
10    // Start is called before the first frame update
11    0 references
    private void Start()
12    {
13        suaraFinish = GetComponent();
14    }
15
16    0 references
    private void OnTriggerEnter2D(Collider2D other)
17    {
18        if (other.gameObject.tag == "Player" && !selesaiLevel)
19        {
20            suaraFinish.Play();
21            selesaiLevel = true;
22            Invoke("LevelSelesai", 1.5f);
23        }
24    }
25
26    0 references
    private void LevelSelesai()
27    {
28        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
29    }
30 }
31
32 }

```

Kode diatas merupakan kode **Finish.cs** yang berfungsi untuk Game agar bisa berpindah Scene ke level selanjutnya dan memberi sound effect apabila Player sudah mencapai bendera Finish atau menyentuh bendera Finish. Terdapat fungsi LevelSelesai() untuk berpindah Scene/Level, lalu ada bool selesaiLevel, agar Sound Effect tidak ter-play terus menerus.

```

6 public class NextScene : MonoBehaviour
7 {
8     // Start is called before the first frame update
9     0 references
    void Start()
10    {
11        Invoke("SceneLanjut", 2f);
12    }
13    0 references
    private void SceneLanjut()
14    {
15        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
16    }
17 }
18
19

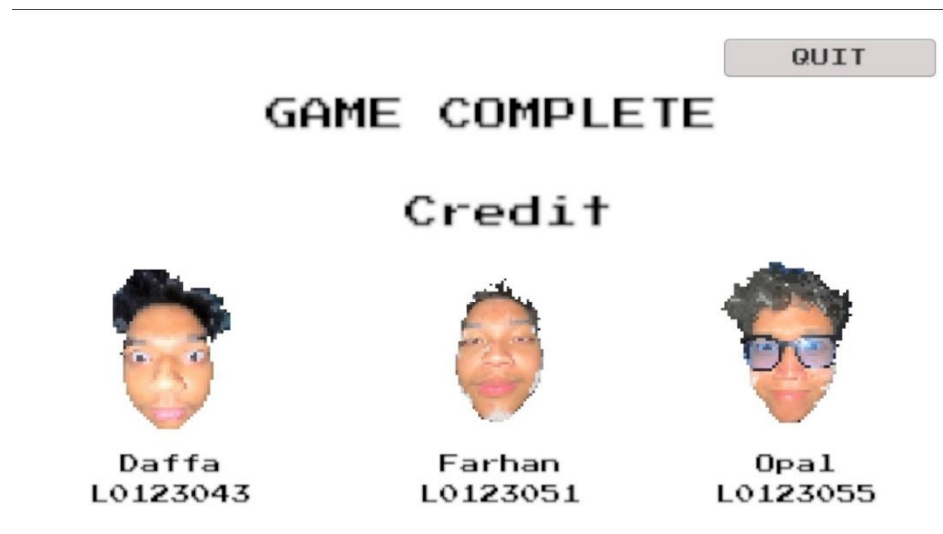
```

LEVEL 1



Pada awal level terdapat transisi, Source Code **NextScene.cs** ini lah yang melanjutkan scene transisi ke scene Level.

Credit Screen



```
5 public class EndMenu : MonoBehaviour
6 {
7     0 references
8     public void Quit()
9     {
10         Application.Quit();
11     }
12 }
```

Kode diatas adalah kode **EndMenu.cs**, kode ini berfungsi pada button KELUAR/EXIT, yang berfungsi untuk keluar dari program atau Game.

Pembagian Tugas Anggota

1. Daffa Hasanah Arkaan (L0123043)
 - a. Membuat Maps per level
 - b. Sound Effect "Level 1" dan "Level 2" (saat pergantian level)
 - c. Membuat Kode
2. Farhan Cahya Permana (L0123051)
 - a. Membuat Kode
 - b. Sound Effect "Aduh!" dan "Hek!" (saat terkena trap/mati dan saat loncat)
 - c. Membuat UI Menu Awal dan Akhir
3. Fawwaz Naufal Maulana (L0123055)
 - a. Membuat Kode
 - b. Sound Effect "Yeee!" (saat mencapai finish/menang)
 - c. Membuat Assets Player