

LAPORAN ANALISIS PEMROGRAMAN GAME

Mata Kuliah: Pemrograman Game

Dosen Pengampu: Affix Mareta, S.Pd., M.Eng.



Bridging Education To The Real World

Disusun oleh:

Phangestin Jen – 31200040

Kelas 5PSI51

Prodi Sistem Informasi

Fakultas Teknologi dan Desain

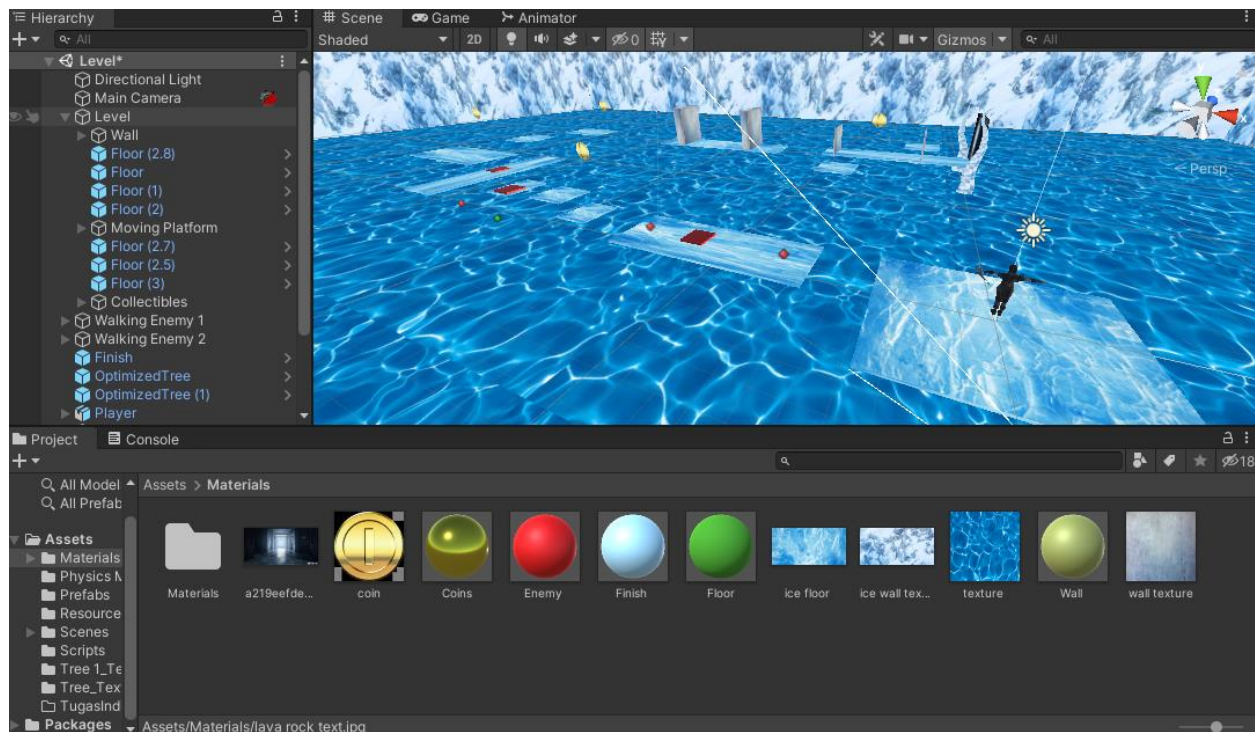
UNIVERSITAS BUNDA MULIA

2022

Concept

Dalam melakukan pembuatan game ini, saya menerapkan metode pengembangan *agile* yang merupakan tahapan pengembangan perangkat lunak dengan bertahap dan dapat berulang (iterasi) yang memiliki keuntungan yaitu tingkat fleksibilitas yang tinggi sehingga dapat menyesuakannya dengan keinginan user. Konsep game yang dibuat ialah game yang cukup sederhana mengikuti konsep game-game terdahulu seperti Mario Bros, dimana player diharuskan untuk melewati rintangan yang ada dan mengumpulkan *item* di setiap perjalanan pada setiap stage untuk mencapai finish.

Berikut ini merupakan gambaran secara keseluruhan dari game tersebut.



Namun walaupun memiliki konsep yang sama dengan game pendahulu Mario Bros, pada game ini menggunakan konsep 3 dimensi yang membuat kita tidak bergerak ke kiri dan kanan saja tetapi dapat bergerak ke berbagai arah dan juga kamera yang sudah diatur menggunakan fitur *Cinemachine* yang tersedia pada Unity sehingga player akan dapat mengatur kamera ke berbagai sisi hanya dengan menggerakkan *mouse* atau kursor. Game KZ ini juga mempunyai *enemy* sebagai rintangan tambahan kepada player yang apabila terkena *enemy* tersebut maka player

akan kalah sama seperti ketika jatuh. Namun *enemy* ini dapat dihilangkan atau dikalahkan oleh player ketika player melompat ke atasnya sehingga *enemy* tersebut akan menghilang dari map.

Konsep dalam permainan KZ Parkour mengambil referensi dari buku “THE ART OF GAME DESIGN” yang mana dalam buku tersebut menjelaskan beberapa konsep yang diterapkan dalam game KZ Parkour diantaranya ialah mekanisme sebuah game yang dapat memberikan tantangan kepada para pemain agar dapat menguji fokus pemain yang mencoba untuk memenangkan setiap rintangan yang ada dengan perolehan waktu yang lebih baik dan skor koin yang dikumpulkan lengkap.

Rintangan tambahan lainnya yang cukup menantang ialah pada game ini memiliki platform dan *wall* atau dinding yang bergerak sehingga dapat membuat player harus berhati-hati dalam mengambil tindakan untuk melewatinya karena platform yang bergerak tersebut ketika dipijaki oleh player, maka player tidak akan ikut terseret secara otomatis bersama dengan platform bergerak tersebut melainkan player juga harus mengikuti pergerakan platform tersebut.

Design

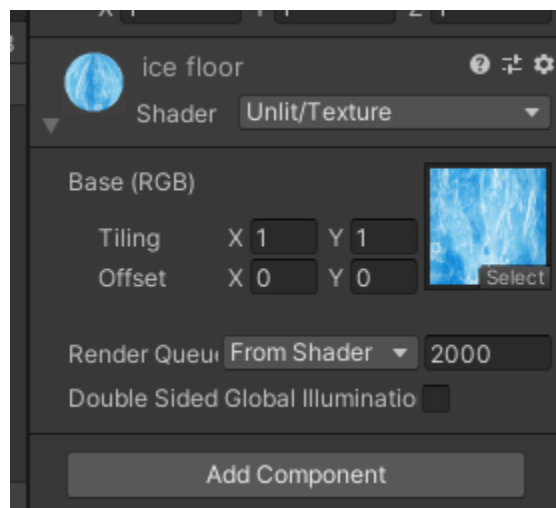
Dalam game KZ, desain yang digunakan terhadap latar tempat atau *background* dari map tersebut mengambil latar tempat lautan es. Dengan tahapan dalam pembuatan desain latar ini ialah:

1. Mencari texture animasi dan gambar dari karakter yang akan dimainkan oleh player
2. Mencari texture gambar untuk background bagian bawah atau *ground* yang akan diisi dengan gambar lautan es.
3. Mencari texture gambar untuk background bagian sisi yang akan diisi dengan gambar texture gunung es
4. Mencari texture gambar untuk background yang akan digunakan untuk pijakan player dengan gambar texture es.
5. Mencari texture gambar untuk background yang akan digunakan untuk *wall* yang bergerak

6. Mencari texture gambar untuk koin yang menjadi fitur tambahan untuk player mengumpulkan *item collect*.
7. Mencari texture gambar untuk game object yang dijadikan finish.

Untuk melakukan desain dari tahapan nomor 2 sampai 7, secara garis besar tahapan yang dilakukan sama, yaitu:

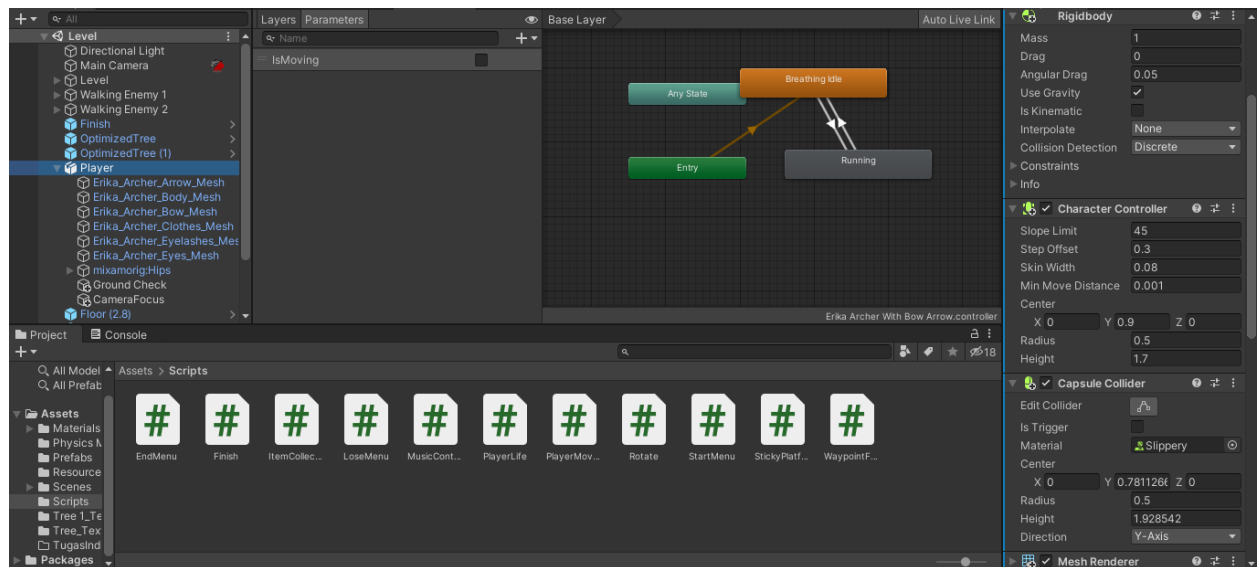
1. Mencari dan mendownload texture yang tersedia di Google.
2. Menentukan tempat simpan dari file unduhan texture
3. Melakukan *drag and drop* gambar texture tersebut kedalam file asset pada Unity
4. Melakukan *drag and drop* gambar texture yang sudah tersimpan dalam file asset pada Unity ke object yang sudah ada
5. Mengatur Shader pada gambar texture yang tersebut dengan memilih Unlit/Texture
6. Mengatur Tiling pada texture tersebut yang tersedia pada pengaturan Base (RGB).



Dan berikut ini merupakan tahapan dari desain karakter beserta animasi bergerak dan *idle* dari karakter yang akan dimainkan oleh player.

1. Mencari dan mengunduh file karakter di website mixamo
2. Mencari dan mengunduh file animasi bergerak yang tersedia di website mixamo

3. Mencari dan mengunduh file animasi *idle* yang tersedia di website mixamo.
4. Menyimpan file unduhan tersebut pada file asset Unity
5. Melakukan *drag and drop* karakter tersebut ke dalam game object yang dijadikan sebagai player
6. Melakukan *drag and drop* animasi *idle* dan bergerak pada game object yang dijadikan sebagai player
7. Menambahkan beberapa komponen pada game object yang dijadikan player tersebut diantaranya Rigidbody, Character Controller, Collider, Animator, serta script untuk bergerak, life, dan item collector.
8. Memberikan pengaturan pada komponen Animator dengan menyambung garis pada *idle* dan *running* seperti di gambar



9. Setelahnya, melakukan coding pada script

```

24 void Start()
25 {
26     rb = GetComponent<Rigidbody>();
27     animator = GetComponent<Animator>();
28     characterController = GetComponent<CharacterController>();
29     originalStepOffset = characterController.stepOffset;
30 }

```

yang berguna ketika game start, maka komponen tersebut akan mengakses animasi yang sudah dilakukan *setting* sehingga karakter memiliki animasi *idle* dan *running* ketika bergerak.

10. Selanjutnya, pada prosedur Update()

```
33 void Update()
34 {
35     float horizontalInput = Input.GetAxis("Horizontal");
36     float verticalInput = Input.GetAxis("Vertical");
37
38     Vector3 movementDirection = new Vector3(horizontalInput, 0, verticalInput);
39     float magnitude = Mathf.Clamp01(movementDirection.magnitude) * speed;
40     movementDirection.Normalize();
41
42     movementDirection = Quaternion.AngleAxis(cameraTransform.rotation.eulerAngles.y, Vector3.up) * movementDirection;
43
44     yspeed += Physics.gravity.y * Time.deltaTime;
45
46     if(characterController.isGrounded){
47         lastGroundedTime = Time.time;
48     }
49     if(Input.GetButtonDown("Jump")){
50         jumpButtonPressedTime = Time.time;
51     }
52
53     if(Time.time - lastGroundedTime <= jumpButtonGracePeriod){
54         characterController.stepOffset = originalStepOffset;
55         yspeed = -0.5f;
56
57         if(Time.time - jumpButtonPressedTime <= jumpButtonGracePeriod){
58             yspeed = jumpSpeed;
59             jumpButtonPressedTime = null;
60             lastGroundedTime = null;
61         }
62     }
63     else {
64         characterController.stepOffset = 0;
65     }
66 }
```

berfungsi sebagai pergerakan karakter yang nantinya dapat bergerak dan melompat dalam pengimplementasiannya dengan menyesuaikan posisi secara otomatis pada transform dari Unity.

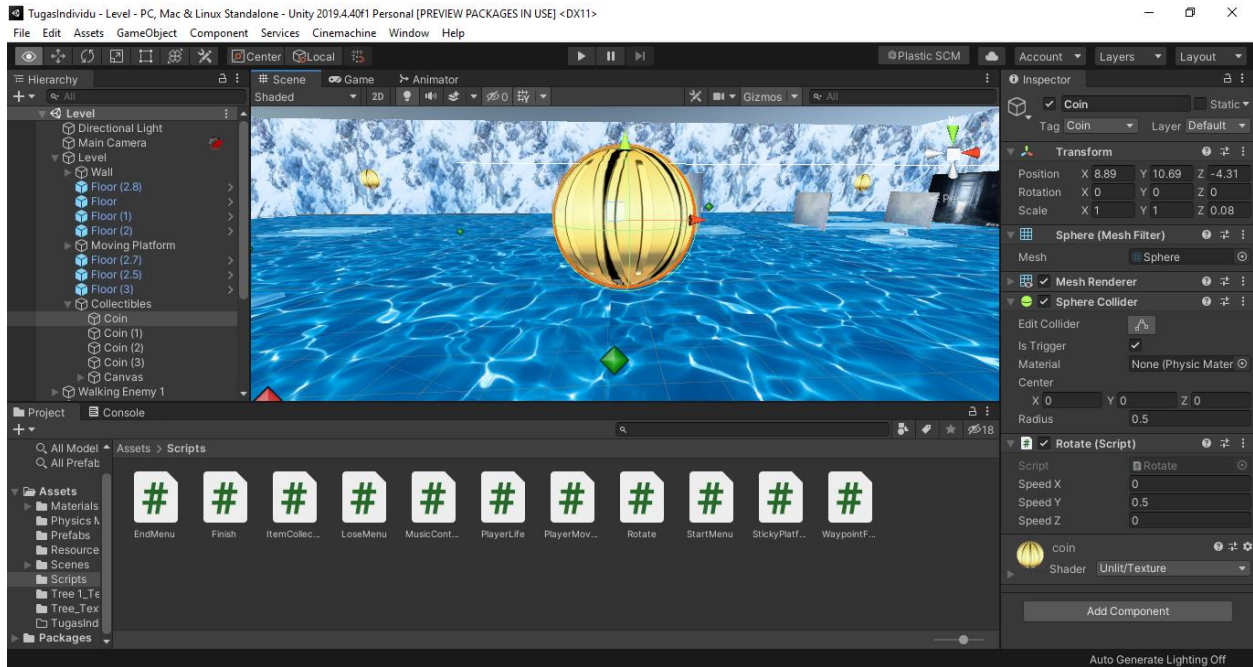
11. Kemudian terdapat percabangan

```
if(movementDirection != Vector3.zero){
    animator.SetBool("IsMoving", true);
    Quaternion toRotation = Quaternion.LookRotation(movementDirection, Vector3.up);
    transform.rotation = Quaternion.RotateTowards(transform.rotation, toRotation, rotationSpeed * Time.deltaTime);
}
else {
    animator.SetBool("IsMoving", false);
}

private void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.CompareTag("Enemy Head"))
    {
        Destroy(collision.transform.parent.gameObject);
    }
}
```

yang berfungsi untuk membaca pergerakan karakter apabila karakter bergerak, maka akan menjalankan pengaturan dari animator untuk *running* dan ketika pergerakan karakter berhenti, maka akan menjalankan pengaturan dari animator untuk *idle*. Dan juga terdapat coding selanjutnya yang berguna untuk ketika player melompat dan menyentuh bagian atas pada *enemy* maka *enemy* tersebut akan menghilang atau berhasil dikalahkan.

Material Collecting



Pada stage game tersebut terdapat koin yang menjadi *item collect* atau fitur tambahan untuk player yang dapat dikumpulkan. Berikut ini merupakan tahapan dalam memberikan game object yang menjadi material collecting.

1. Menambahkan game object pada unity dengan bentuk sphere
2. Membuat script untuk rotasi agar dapat berputar koin tersebut ketika game dijalankan

```
5 public class Rotate : MonoBehaviour
6 {
7     [SerializeField] float speedX;
8     [SerializeField] float speedY;
9     [SerializeField] float speedZ;
10    // Update is called once per frame
11    void Update ()
12    {
13        transform.Rotate(360 * speedX * Time.deltaTime, 360 * speedY * Time.deltaTime, 360 * speedZ * Time.deltaTime);
14    }
15 }
16
```

3. Memberi game object yang menjadi koin tersebut dengan script yang telah dibuat



4. Membuat script untuk game object yang dijadikan sebagai player dengan nama script Item Collector

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class ItemCollector : MonoBehaviour
7  {
8      int coins = 0;
9
10     [SerializeField] Text coinsText;
11
12     private void OnTriggerEnter(Collider other) {
13         if (other.gameObject.CompareTag("Coin")) {
14             Destroy(other.gameObject);
15             coins++;
16             coinsText.text = "Coins : " + coins;
17         }
18     }
19
20 }
21
22
```

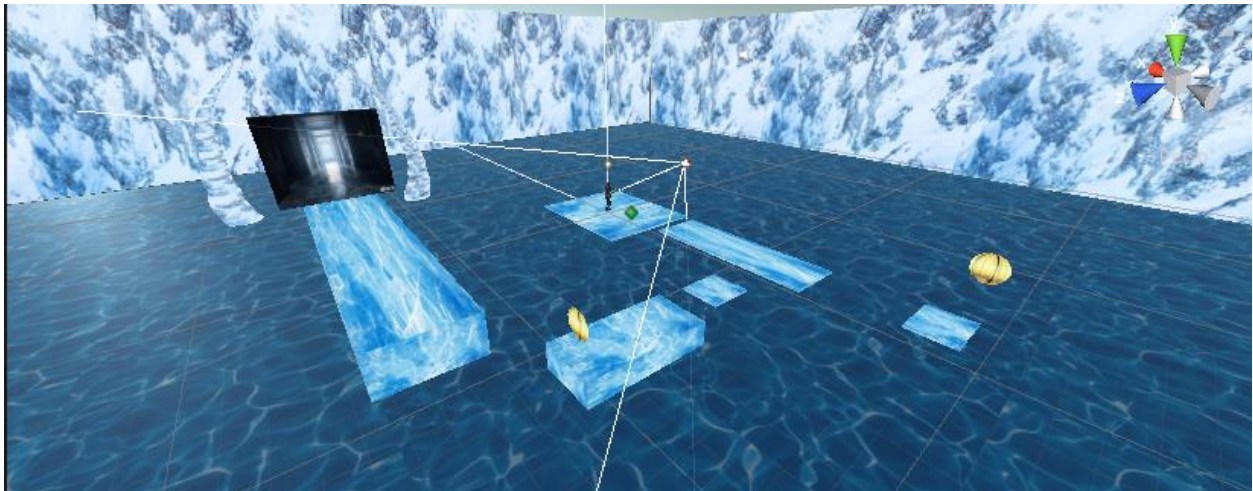
Yang berguna ketika player menyentuh koin tersebut, maka koin akan menghilang dan jumlah koin yang dikumpulkan akan bertambah yang dapat dilihat pada bagian atas kiri layar ketika dimainkan.



5. Memberikan gambar texture koin dengan tahapan yang sudah dijelaskan pada bagian desain dengan melakukan *drag and drop*.

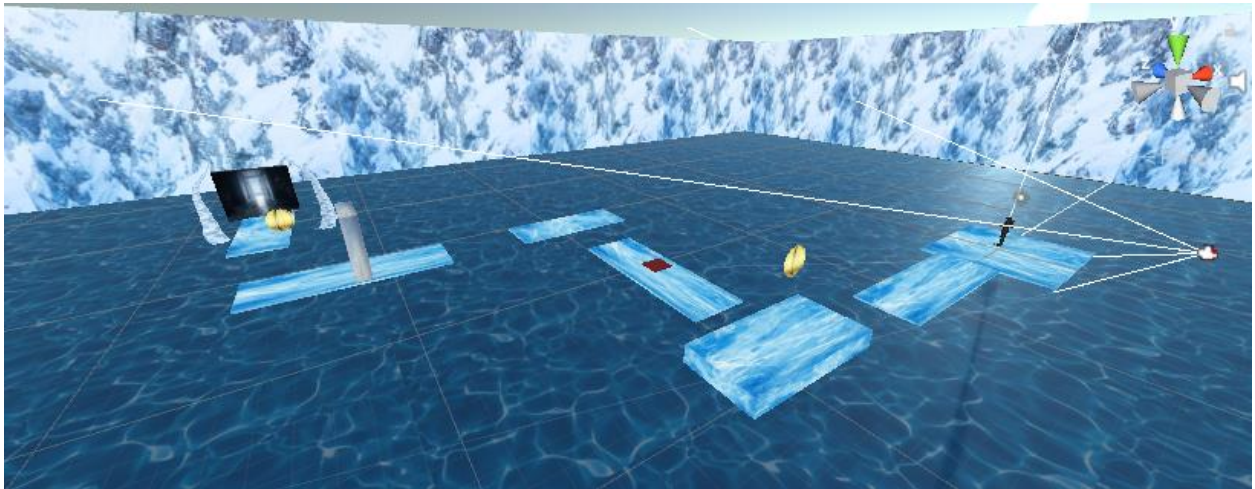
Assembly

Dalam tahapan pembuatan stage, terdapat 3 level dari stage yang dibuat dengan setiap level memiliki perkembangan tingkat kesulitan secara bertahap yaitu pada level 1 rintangan yang ada masih memiliki tingkat kesulitan yang mudah yaitu hanya terdapat platform yang bergerak saja. Berikut ini keseluruhan gambaran dari level 1.



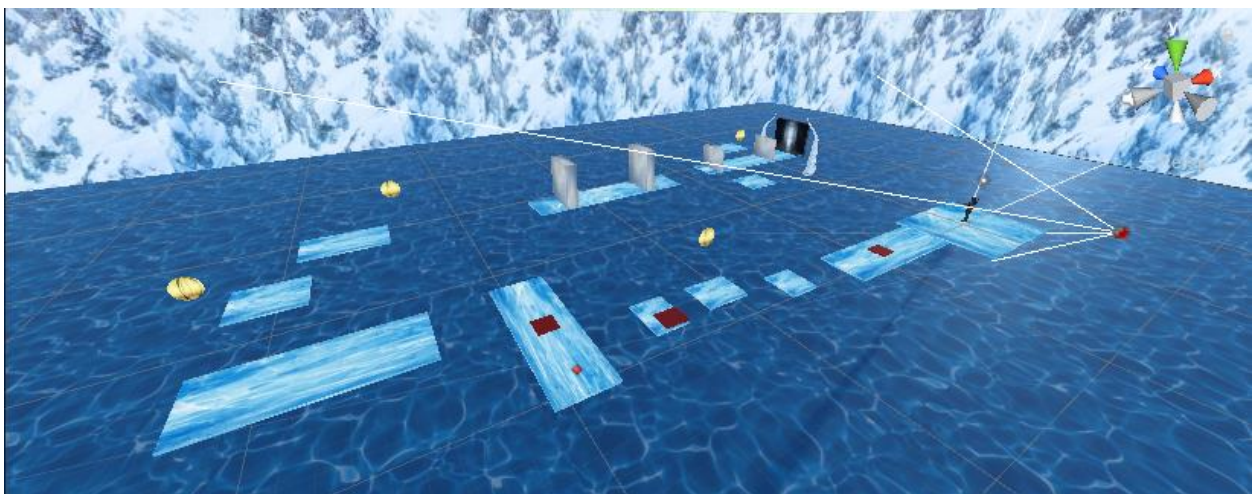
Karena level 1 sebagian besar hanyalah ditujukan untuk pengenalan terhadap permainan game secara keseluruhan, maka tentunya level 1 dibuat sesederhana mungkin agar pemain dapat mempelajari mekanisme dari permainan dengan mudah dan dapat menarik kesimpulan secara cepat untuk garis besar cara memainkannya.

Pada tingkatan lanjutnya juga terdapat level 2 yang sudah mulai hadir tantangan yang baru dengan adanya *enemy* dan juga *wall* yang bergerak yang dapat menggeser posisi dari pemain hingga jatuh apabila terkena *wall* tersebut. Berikut ini merupakan gambar dari keseluruhan rintangan yang ada pada level 2.



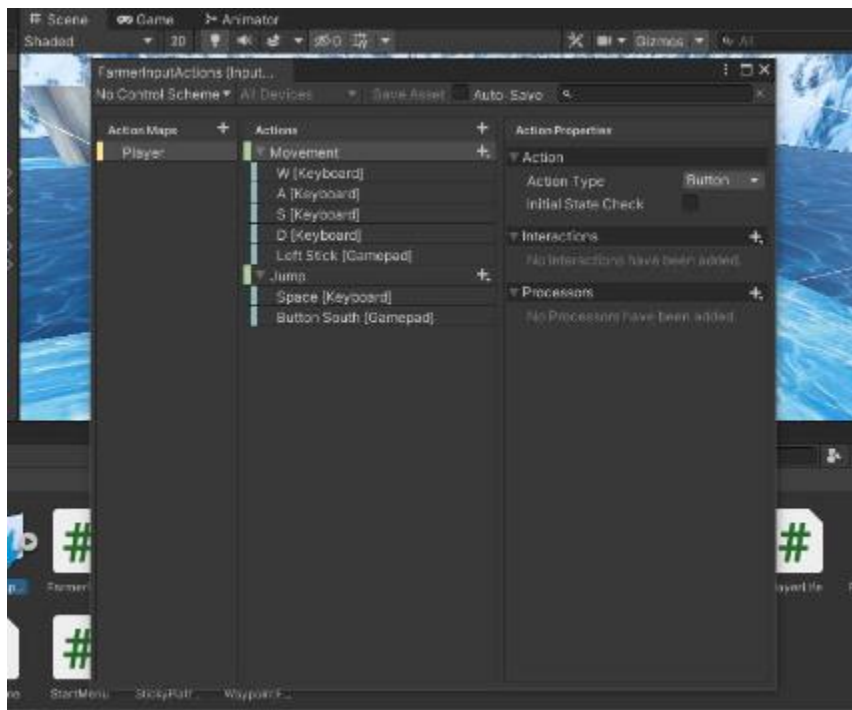
Pada level 2 ini tidak ditemukan adanya rintangan *platform* yang bergerak karena memang ditujukan untuk pengenalan terhadap rintangan yang baru dibuat sehingga nantinya ketika pemain sudah memasuki level selanjutnya, keseluruhan dari rintangan yang ada akan ditampilkan menjadi satu scene.

Pada tingkatan level selanjutnya yaitu level 3, keseluruhan rintangan pun ditampilkan sehingga pada level 3 menjadi level yang cukup sulit untuk para pemain. Dari hal tersebut maka dapat terpenuhi konsep dari mekanisme yang terdapat dalam buku "THE ART OF GAME DESIGN". Berikut ini merupakan tampilan untuk penggambaran level 3.



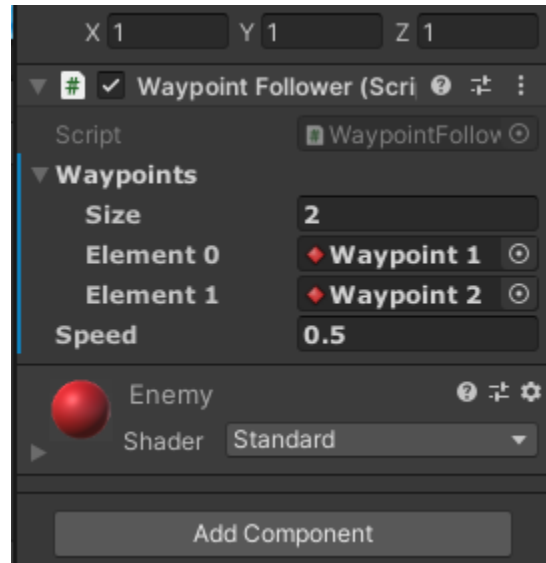
Jumlah rintangan yang terdapat pada level 3 ini diantaranya terdapat 3 *enemy*, 2 *wall* yang bergerak, serta 3 buah platform yang bergerak yang mengharuskan pemain untuk fokus terhadap platform bergerak tersebut dikarenakan karakter pemain yang tidak dapat mengikuti pergerakan platform tersebut sehingga pemain harus menyesuaikannya. Tentunya dari hal tersebut akan membuat pengalaman dari pemain akan meningkat sehingga dalam perkembangan permainan selanjutnya hingga level yang cukup tinggi maka akan dibuat rintangan platform bergerak tersebut dengan kecepatan yang semakin meningkat seiring dengan perkembangan pengalaman pemain menghadapi rintangan tersebut.

Dalam tahapan pembuatan permainan ini juga ditambahkan fitur agar pemain dapat memainkannya dengan menggunakan *joystick* ataupun *controller* yang mungkin dapat memudahkan para pemain untuk bergerak tanpa menggunakan W, A, S, D. Berikut ini merupakan tampilan terhadap fitur tersebut.



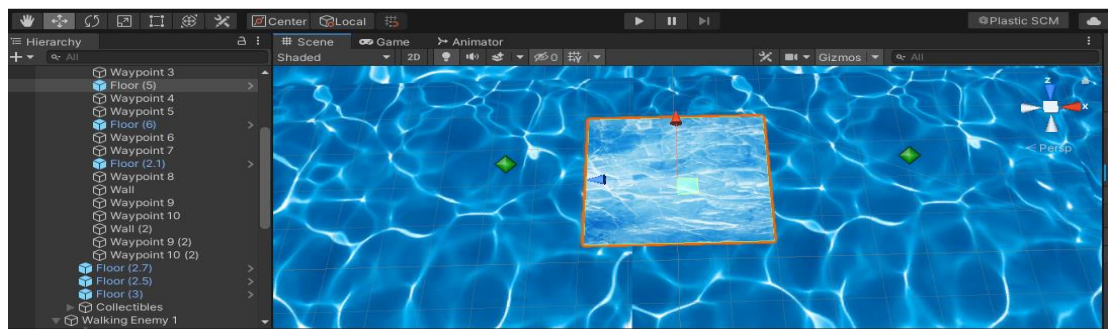
Telah dilakukan *setting* terhadap *movement* dari karakter pemain sehingga pemain dapat menggunakan tombol dari *left stick controller* untuk bergerak lebih leluasa tanpa menggunakan *keyboard* W,A,S,D. Serta pemain dapat melompat dengan menggunakan *controller* dengan menekan tombol bagian selatan yang terdapat pada *controller*.

Dalam tahapan pembuatan stage juga terdapat beberapa bagian dari platform dan dinding yang bergerak dengan menggunakan game object tambahan yang diberikan script codingan untuk dapat mengulangi pergerakannya dengan nama script Waypoint Follower.



Berikut ini merupakan tahapan dalam pembuatannya:

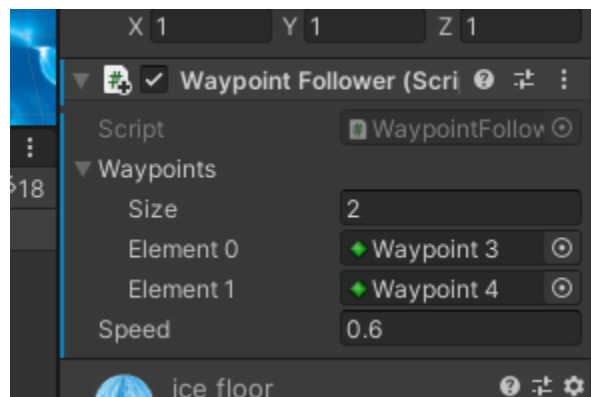
1. Menambahkan 2 object baru dan mengubah bentuk ikonnya menjadi bentuk *diamond* serta menyesuaikan posisi transformnya agar sama seperti platform atau object lainnya yang ingin dijadikan bergerak.
2. Setelah posisi transform telah sama seperti object yang ingin digerakkan, maka object yang pertama dipindahkan dengan menggunakan Move Tool yang tersedia pada Unity agar transform posisi yang berubah dapat lurus ke sisi kiri dan kanan (sisi kiri object ke 1, sisi kanan object ke 2) seperti pada gambar berikut ini.



3. Membuat script Waypoint Follower agar platform dapat bergerak ke sisi dan kanan sesuai penempatan object 1 dan 2 secara lurus.

```
7 [SerializeField] GameObject[] waypoints;  
8 int currentWaypointIndex = 0;  
9  
10 [SerializeField] float speed = 1f;  
11  
12 void Update()  
13 {  
14     if (Vector3.Distance(transform.position, waypoints[currentWaypointIndex].transform.position) < .1f)  
15     {  
16         currentWaypointIndex++;  
17         if (currentWaypointIndex >= waypoints.Length)  
18         {  
19             currentWaypointIndex = 0;  
20         }  
21     }  
22     transform.position = Vector3.MoveTowards(transform.position, waypoints[currentWaypointIndex].transform.position, speed * Time.deltaTime);  
23 }  
24  
25  
26
```

4. Ketika script telah selesai dibuat, maka berikan script tersebut ke object yang ingin dijadikan dapat bergerak ke 2 sisi tersebut dan melakukan *drag and drop* kedua object yang menjadi batas pergerakan platform tersebut sehingga menjadi elemen 1 dan 2 seperti pada gambar.



Secara garis besar, dalam pembuatan stage tersebut untuk setiap object yang bergerak menggunakan cara yang sama seperti *enemy*, *platform*, dan *wall* yang menjadi rintangan untuk player sehingga mengharuskan untuk bergerak secara hati-hati. Pada player juga diberikan script *PlayerLife* yang berfungsi ketika player gagal atau tidak berhasil melompati dan memijak bagian atas *enemy* maka akan kalah. Berikut script *PlayerLife*.

```

7 public class PlayerLife : MonoBehaviour
8 {
9     bool dead = false;
10
11     private void Update() {
12         if (transform.position.y < -1f && !dead) {
13             Die();
14             if (dead == true) {
15                 SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 0);
16             }
17         }
18     }
19
20     private void OnCollisionEnter(Collision collision)
21     {
22         if (collision.gameObject.CompareTag("Enemy Body"))
23         {
24             GetComponent<MeshRenderer>().enabled = false;
25             GetComponent<Rigidbody>().isKinematic = true;
26             GetComponent<PlayerMovement>().enabled = false;
27             Die();
28         }
29     }
30
31
32     void Die() {
33         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 0);
34         dead = true;
35     }
36
37     void ReloadLevel() {
38         SceneManager.LoadScene(SceneManager.GetActiveScene().name);
39     }
40
41 }
42

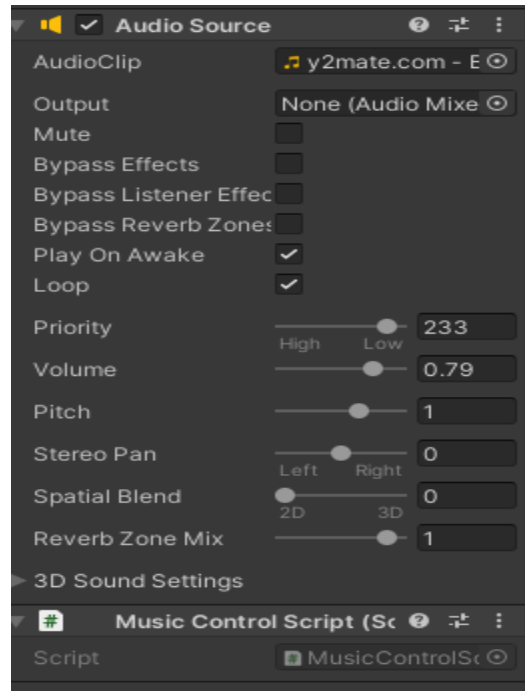
```

Ketika nilai dari Boolean variable `dead = true` maka scene akan langsung dialihkan ke bagian scene yang sama sehingga apabila player telah kalah pada level yang terkait, maka akan langsung melakukan pengulangan otomatis.

Dalam game KZ, juga ditambahkan lagu atau *backsound* yang dapat menambah keseruan dari game tersebut. Berikut ini merupakan tahapan dalam penambahan sound tersebut:

1. Mendownload lagu yang akan dijadikan *backsound*
2. Menambahkan game object baru dengan nama Music Control
3. Melakukan *drag and drop* lagu tersebut ke dalam file asset unity
4. Menambahkan component Audio Source pada game object Music Control

5. Melakukan *drag and drop* kembali dengan lagu yang dipilih ke dalam source dari komponen audio source seperti pada gambar.



6. Memberikan script pada object tersebut agar music dapat berjalan ketika game dijalankan. Berikut ini merupakan script dari music tersebut.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class MusicControlScript : MonoBehaviour
6 {
7     public static MusicControlScript instance;
8
9     private void Awake() {
10         DontDestroyOnLoad(this.gameObject);
11
12         if(instance == null) {
13             instance = this;
14         }
15         else {
16             Destroy(gameObject);
17         }
18     }
19 }
20
```

Testing

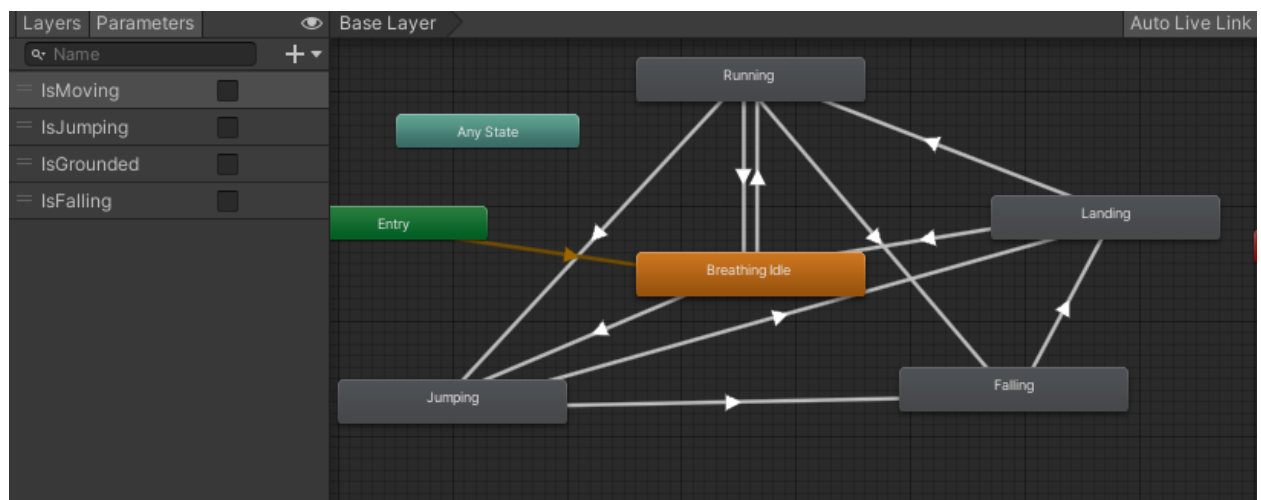
Dalam tahapan testing, telah dilakukan kepada beberapa orang untuk menjalankan game KZ dan berikut ini beberapa respon game ketika pertama kali dijalankan memiliki beberapa kekurangan diantaranya:

1. Tidak dapat mengetahui arah tujuan dari stage tersebut dikarenakan kamera yang tidak dapat berputar. Sehingga dalam tahapan yang dilakukan sekarang, telah ditambahkan object dengan menggunakan fitur Cinemachine yang terdapat pada Unity agar kamera dapat berputar 360 derajat sehingga player dapat menyesuaikan posisi kamera dengan rintangan yang ada di depannya.
2. Rintangan seperti *enemy*, platform, dan *wall* yang bergerak sangat sulit untuk dilewati untuk para pemain yang baru mencoba atau memasuki industri dunia game. Sehingga dalam perancangan game sekarang dilakukan pengurangan pada kecepatan rintangan yang bergerak, dan mengurangi beberapa jumlah object yang dapat bergerak sehingga menjadi lebih *user friendly*.
3. Tidak adanya background dan masih menggunakan background default juga menjadi respon sehingga *value* dari game itu sendiri dapat berkurang. Pada perancangan game sekarang telah dilakukan penambahan background serta musik sehingga menjadikan game lebih berwarna dan semakin menantang dengan musik yang diberikan.
4. Kurangnya fitur *item collect* sehingga player hanya berfokus untuk rintangan yang ada saja. Pada perancangan game sekarang telah dilakukan penambahan koin sebagai item collect yang dapat dikumpulkan oleh para player.

Setelah dilakukan beberapa perbaikan dan pengoptimasian, maka berikut ini merupakan tahapan yang dijadikan saran untuk para pemain agar dapat bermain dengan tahapan yang sudah ditentukan dan direncanakan:

1. Saat memulai game, diharapkan setiap player untuk mengatur dan mencari posisi dari kamera terlebih dahulu karena ketika game awal dimulai, kamera akan melakukan *zoom in* pada karakter yang dapat membuat beberapa player tidak nyaman. Untuk itu, player dapat melakukan *zoom out* pada kamera dan mengatur peletakan posisinya agar terlihat nyaman dan dapat melihat rintangan berikutnya.

2. Player disarankan untuk tetap bersabar ketika melewati rintangan yang ada dan ketika ingin mengubah posisi kamera, disarankan untuk mengubahnya ketika posisi platform diam dan aman untuk diganti posisi kameranya.
3. Player tidak diwajibkan untuk mengumpulkan koin yang ada karena koin tersebut hanya menjadi fitur tambahan saja dan tidak menjadi kunci dari kemenangan stage tersebut.
4. Player dapat bergerak menggunakan tombol W,A,S,D ataupun menggunakan anak panah pada keyboard dengan menyesuaikan kamera pada game serta SPACE untuk melompat. Player juga dapat bergerak dengan menyerong kiri kanan dengan menekan dua tombol secara bersamaan.
5. Animasi bergerak karakter yang ketika sebelum dilakukan pengujian terhadap para pemain hanyalah mempunyai animasi *idle*, dan *running* saja. Sehingga dalam perkembangannya, kini ditambahkan animasi ketika melakukan melompat ke atas, melayang di udara, dan juga animasi ketika mendarat. Berikut ini keseluruhan dari penggambaran animasi yang telah dikembangkan.



6. Setelah dilakukan testing, terdapat fitur tambahan baru dalam permainan yaitu fitur melakukan *pause* game yang bertujuan untuk memberikan kesempatan untuk pemain agar dapat beristirahat sejenak atau melakukan kegiatan lainnya sehingga ketika mereka selesai dalam melakukan kegiatannya, dapat melanjutkan kembali permainannya. Berikut ini merupakan tampilan dari *pause* game yang dilakukan ketika pemain menekan tombol ESC.



Pada menu ini pemain dapat keluar dari permainan dengan menekan tombol *quit game*, ataupun melanjutkan permainan dengan menekan tombol *resume* sehingga permainan dapat berjalan kembali.

Distribution

Dalam tahapan pendistribusian game, KZ game ini dapat dijalankan dengan spesifikasi yang minim sehingga cukup bersahabat dengan spesifikasi komputer yang *low*. Karena masih merupakan tahap awal dimana hanya terdapat 1 level saja, direncanakan hingga tahap pendistribusian setidaknya memiliki 5 level dan dalam pemasaran tidak akan dilaksanakan tahapan CBT atau *Close Beta Test* terlebih dahulu mengingat game KZ merupakan game dengan konsep yang sederhana sehingga mungkin kebanyakan para pemain akan mengetahui bahwa KZ game mempunyai konsep yang sama dengan game Mario Bros.

Dalam pemasarannya, akan dipasarkan melalui platform seperti steam dan lainnya yang menjadi platform berbagai game yang baru di rilis dengan harga yang gratis karena masih tahap pengembangan dan masih minimnya fitur-fitur.

Dikarenakan besarnya jumlah size dari game yang dibuat, berikut ini merupakan link menuju Google Drive untuk aplikasi game dan video singkat dari *gameplay*.

https://drive.google.com/drive/folders/1PO9Z6H3UpcagoTs8wPEp2imiqteQgCa3?usp=share_link

***Untuk file TugasIndividu merupakan file keseluruhan dari project game (seluruh file yang terdapat dalam Unity termasuk komponen pembentuknya)**

***Untuk file APP merupakan file yang berisi aplikasi tanpa seluruh komponen project Unity yang dibuat.**

***File ZIP Keseluruhan merupakan file yang berisi semua file yang ada kecuali file TugasIndividu (keseluruhan project yang ada pada Unity)**