



TUGAS PERTEMUAN: 8

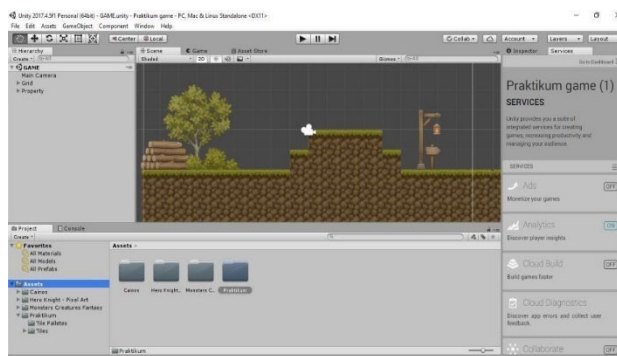
Camera & Character Movement

NIM	:	1818050
Nama	:	Adji Bayu Pamungkas
Kelas	:	F
Asisten Lab	:	
Baju Adat	:	Pakaian Adat Dayak Ngaju
Referensi	:	https://id.pinterest.com/pin/9499849208264490/

8.1 Tugas 1 : Camera & Character Movement

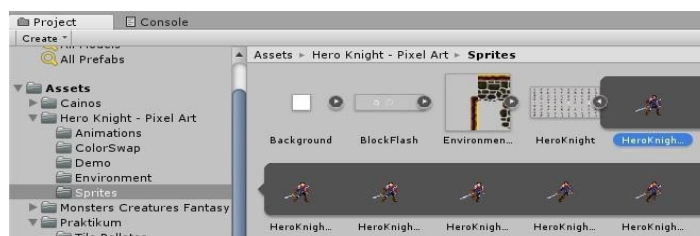
A. Membuat Pergerakan Layer

1. Buka *file project Unity* sebelumnya pada BAB 7 untuk digunakan kembali Seperti pada Gambar 8.3 dibawah ini



Gambar 8.1 Tampilan dari Buka *Project Unity* sebelumnya

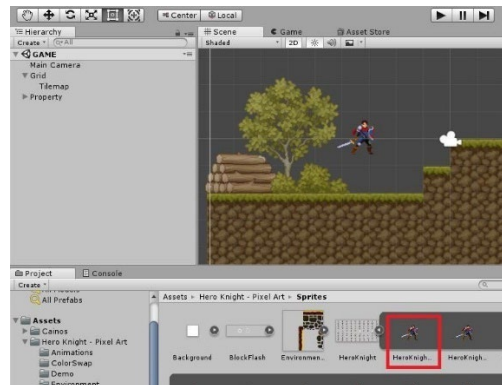
2. Cari *folder Hero Knight* → *Pixel Art* → *Sprites*, Kemudian pada *file Hero Knight*, klik bagian panah kecil dibagian kanan file tersebut, maka akan muncul banyak *file sprite* gambar. Seperti pada Gambar 8.4 dibawah ini



Gambar 8.2 Tampilan dari Membuka *Folder Sprites*

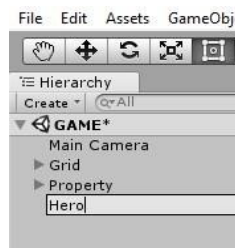


3. Pilih HeroKnight_0 seperti gambar dibawah ini, kemudian *drag* dan *drop* kedalam halaman kerja Seperti pada Gambar 8.5 dibawah ini



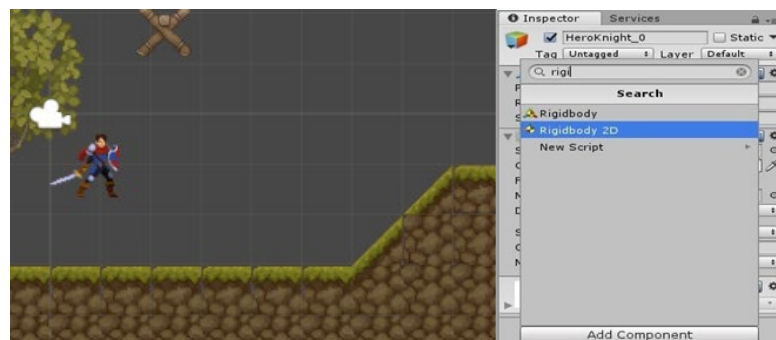
Gambar 8.3 Tampilan dari Memilih *HeroKnight*

4. Pada *Hierarchy*, ganti nama “HeroKnight_0” menjadi “Hero”, caranya klik kanan pada HeroKnight_0, pilih *Rename* Seperti pada Gambar 8.6 dibawah ini



Gambar 8.4 Tampilan dari Merubah Nama Hero

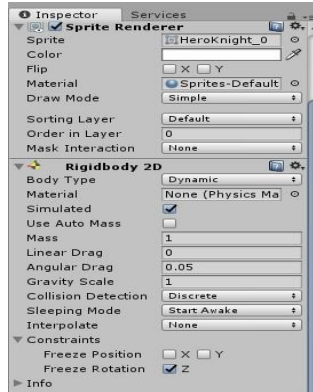
5. Klik karakter tersebut, pergi *Inspector*, klik *Add Component*, dan cari komponen *RigidBody 2D*. *RigidBody* ini berfungsi sebagai pemberian sifat fisik pada karakter. Seperti pada Gambar 8.7 dibawah ini



Gambar 8.5 Tampilan dari Membuka *Rigdnody 2D*

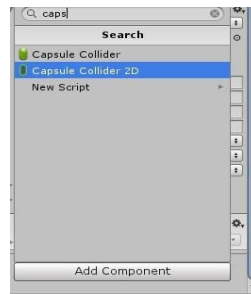


6. Pada properties *Rigidbody 2D*, klik *Constraint*, dan centang pada bagian *Z*, *Freeze Rotation*, kotak tersebut dicentang agar saat karakter menginjak tanah tidak terguling Seperti pada Gambar 8.8 dibawah ini



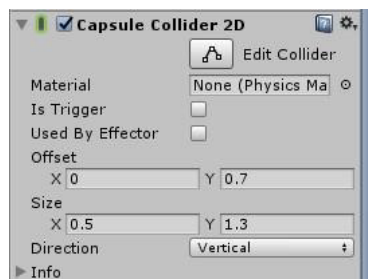
Gambar 8.6 Tampilan dari *Properties Inspector*

7. Tambahkan lagi satu komponen bernama *Capsule Collider 2D*. Collider adalah pembatas antar objek, jadi jika pada objek A dan objek B terdapat *collider* maka satu sama lain tidak dapat saling menembus, jika hanya salah satu memiliki *collider* maka dapat menembus anatar objek. Seperti pada Gambar 8.9 dibawah ini



Gambar 8.7 Tampilan dari *Capsule Collider 2D*

8. Pada *Properties Capsule Collider 2D*, *Input* nilai *Offset* dan *Size* Seperti pada Gambar 8.10 dibawah ini



Gambar 8.8 Tampilan dari *Input Nilai Capsule Collider 2D*

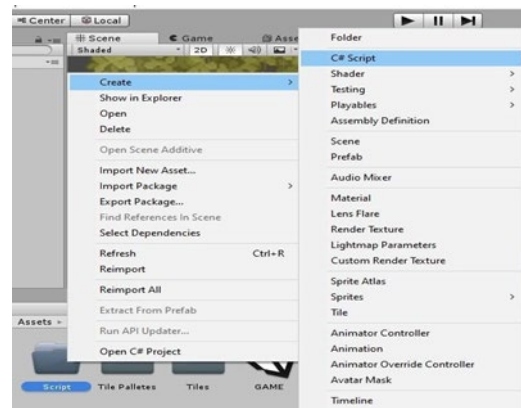


9. Buat *folder* baru bernama “Script” di dalam folder praktikum Seperti pada Gambar 8.11 dibawah ini



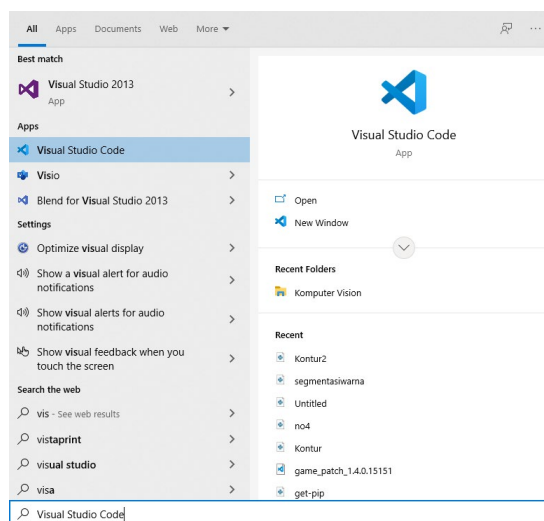
Gambar 8.9 Tampilan dari Membuat *Folder Script*

10. Buat *file Script* baru, caranya klik kanan *folder Script*, pilih *Create* → *C# Script* dan namai *file script* tersebut dengan “Hero” Seperti pada Gambar 8.12 dibawah ini



Gambar 8.12 Tampilan dari Membuat *File Script*

11. Klik 2× *file script* tersebut, maka akan daialihkan ke *Visual Studio Code* Seperti pada Gambar 8.13 dibawah ini



Gambar 8.10 Tampilan dari Aplikasi *Visual Studio Code*



12. Masukkan *source code* dibawah ini, dan untuk mengganti Hero pada *public class Hero* dengan nama yang sama seperti nama file nya

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine; public
class Hero : MonoBehaviour
{
    Rigidbody2D rb;
    [SerializeField] float speed =
    1; float horizontoalValue;
    bool facingRight; void Awake()
    {
        rb =
        GetComponent<Rigidbody2D>();
    } void
    Update()
    {
        horizontoalValue =
        Input.GetAxisRaw("Horizontal");
    } void
    FixedUpdate()
    {
        Move(horizontoalValue);
    } void Move(float
    dir)
    {
        #region gerak kanan kiri
        float xVal = dir * speed * 100 * Time.fixedDeltaTime;
        Vector2 targetVelocity = new Vector2(xVal,
        rb.velocity.y); rb.velocity = targetVelocity; if
        (facingRight && dir < 0)
        {
            transform.localScale = new Vector3(-1, 1, 1);
            facingRight = false;
        }
        else if (!facingRight && dir > 0)
        {
            transform.localScale = new Vector3(1, 1, 1);
            facingRight = true;
        }
        #endregion
    }
}
```



13. *Play Project*-nya, uji karakter dengan bergerak ke kanan dan ke kiri dengan menekan arah kanan kiri di *keyboard* Seperti pada Gambar 8.14 dibawah ini



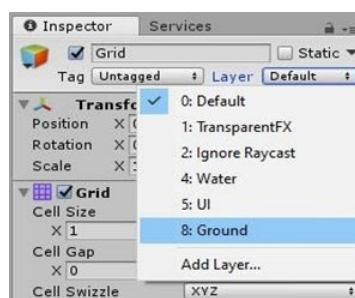
Gambar 8.11 Tampilan hasil dari Uji Coba Karakter

14. Untuk membuat *Ground Check*, klik *Grid* pada *Hierarchy*, pergi ke *Inspector*, pilih *Layer*, klik *Add Layer...* Isikan *User Layer 8* dengan nama *Ground* Seperti pada Gambar 8.15 dibawah ini



Gambar 8.12 Tampilan dari Membuat *Ground*

15. Ganti *Layer Default* menjadi *Ground*, klik *Yes*, dan *change children* Seperti pada Gambar 8.16 dibawah ini



Gambar 8.13 Tampilan dari Merubah Layer Menjadi *Ground*



16. Klik kanan *Hero* kemudian pilih *Create Empty* dan ubah namanya menjadi *GroundCheck* Seperti pada Gambar 8.17 dibawah ini



Gambar 8.14 Tampilan dari Membuat *GroundCheck*

17. Klik *GroundCheck*, kemudian gunakan *Move Tool* untuk menggeser panah berwarna hijau-merah pada objek karakter Seperti pada Gambar 8.18 dibawah ini



Gambar 8.15 ampilan dari Alat *Move Tool*

18. Kembali ke visual studio, tambahkan baris kode

```
[SerializeField] Transform groundcheckCollider;  
[SerializeField] LayerMask groundLayer;    const float  
groundCheckRadius = 0.2f;  
[SerializeField] bool isGrounded;
```

19. Buat void baru dibawah *void FixedUpdate*

```
void GroundCheck()  
{  
    isGrounded = false;  
    Collider2D[] colliders =  
Physics2D.OverlapCircleAll(groundcheckCollider.position,  
    groundCheckRadius, groundLayer);  
    if  
        (colliders.Length > 0)  
isGrounded = true;  
}
```

20. Untuk membuat sebuah mekanisme melompat, tambahkan baris kode seperti dibawah ini

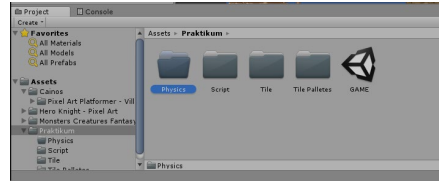


```
[SerializeField] float jumpPower = 100;  
Bool jump;
```

21. Tambahkan kode dibawah ini di dalam void update

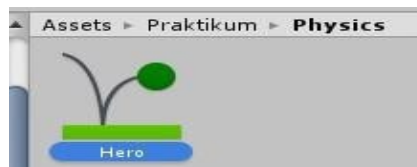
```
if  
(Input.GetButtonDown("Jump" ))          jump = true;  
else if  
(Input.GetButtonUp("Jump"))          jump = false:
```

22. Buat *folder* baru bernama “Physics” dalam folder Praktikum Seperti pada Gambar 8.20 dibawah ini



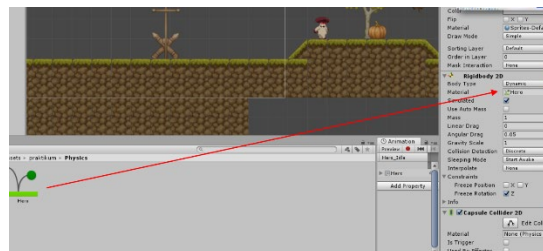
Gambar 8.16 Tampilan dari Membuat Folder Physics

23. Klik kanan folder “Physics” pilih *Create* → *Physics Material 2D* dan beri nama “Hero” Seperti pada Gambar 8.21 dibawah ini



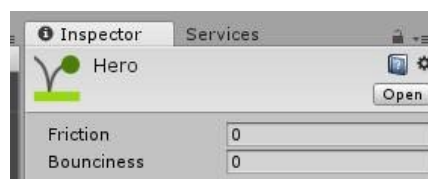
Gambar 8.17 Tampilan dari Membuat *Physics Material*

24. Klik objek Karakter, kemudian *drag file Physics 2D* tadi ke dalam *Material* yang ada pada *Rigidbody 2D* Seperti pada Gambar 8.22 dibawah ini



Gambar 8.18 Tampilan dari *Rigidbody 2D*

25. Klik file tersebut, dan pergi ke *Inspector*, ubah nilai *Friction* dan *Bounciness* menjadi 0 Seperti pada Gambar 8.23 dibawah ini



Gambar 8.19 Tampilan dari Merubah Nilai Pada Inspector



26. Tekan *Play* untuk mencoba hasilnya, karakter sekarang sudah bisa bergerak dan melompat Seperti pada Gambar 8.24 dibawah ini



Gambar 8.20 Tampilan Hasil *Running Project*-nya