

COURSE OUTLINE

GAME PROGRAMMING

Oleh: Abas Setiawan

Contents (DAFTAR ISI)

Pendahuluan	1
Persiapan Asset Game	1
Game Background.....	1
Game Player.....	2
Game Obstacles	4
Scoring System.....	7
Latihan (Optional)	7
REFERENSI	7
Mekanisme Pengumpulan Praktikum Mingguan.....	7

Pendahuluan

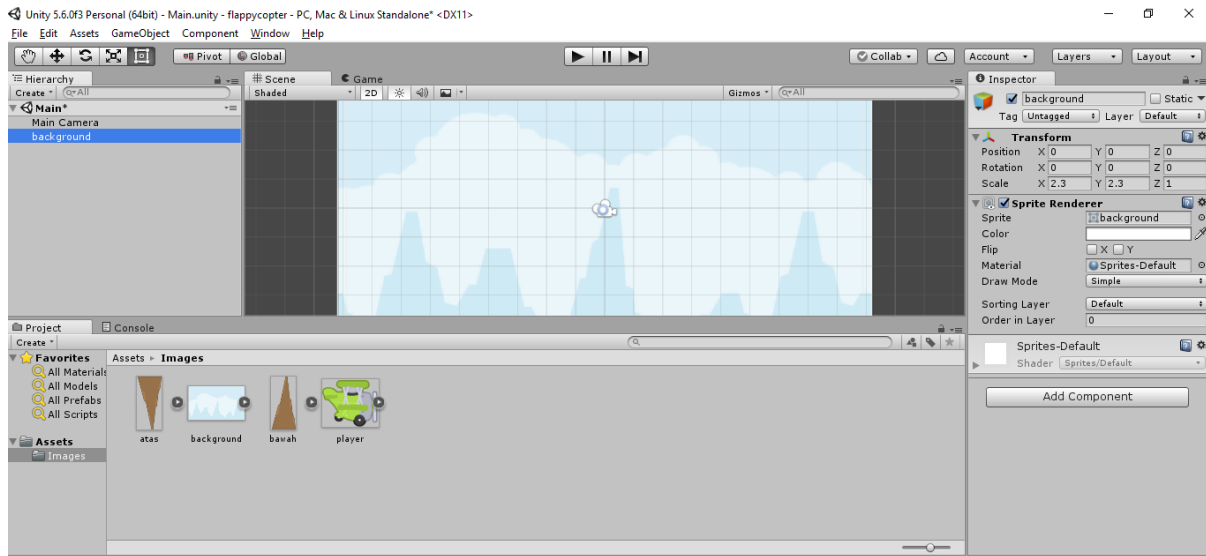
Pada praktikum ke-4 ini, kita akan mencoba membuat project flappy bird 2D. Sebagian besar praktikum ini diambil dari materi perkuliahan Bapak Zikky <http://zikky.lecturer.pens.ac.id/> begitu juga dengan asset game yang digunakan. **Buat project baru 2D dengan nama “flappycopter”, atur aspect ratio 16:9 pada game tab.**

Persiapan Asset Game

Terdapat 4 file gambar yang dapat ditambahkan ke project view, yaitu: atas.png, bawah.png, background.png, dan player.png. Buat folder Images, Prefabs, dan Scripts pada project window. Masukkan ke-empat file gambar tersebut diatas pada folder Images.

Game Background

Masukkan gambar background.png pada scene view dan jadikan background game flappycopter ini. Letakkan game object background pada posisi dan rotasi pada titik (0,0,0). Rubah scale X dan Y pada Transform Inspector dari object background dengan nilai 2.3. Tampilan background akan seperti gambar berikut:

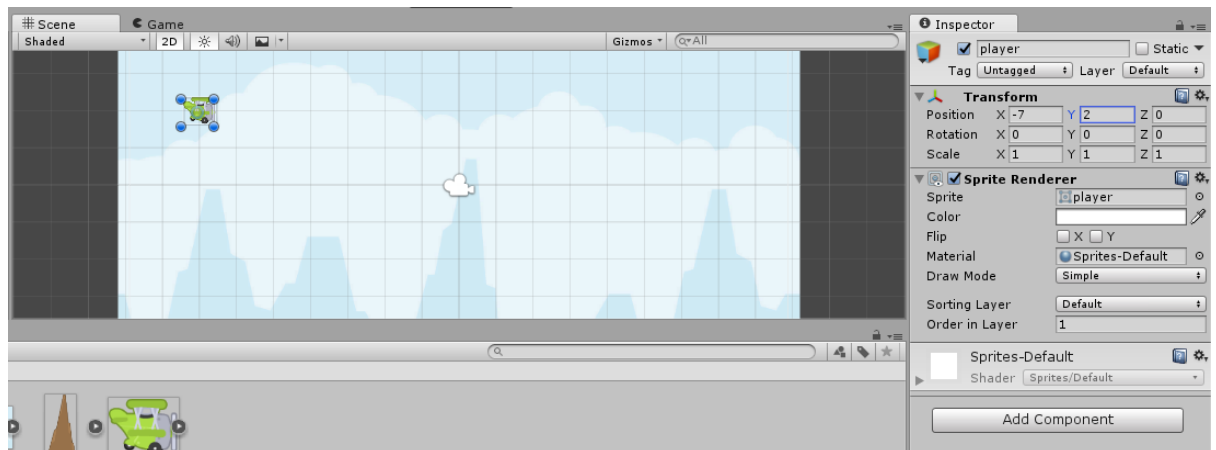


Pastikan pada property Inspector sprite renderer order in layer bernilai 0. Order in layer ini penting untuk meletakkan object pada lapisan scene background atau foreground. Semakin kecil order in layer, maka object akan dapat ditumpuk oleh objek dengan order in layer yang lebih tinggi. Jangan lupa untuk menyimpan scene dengan nama “Main”.

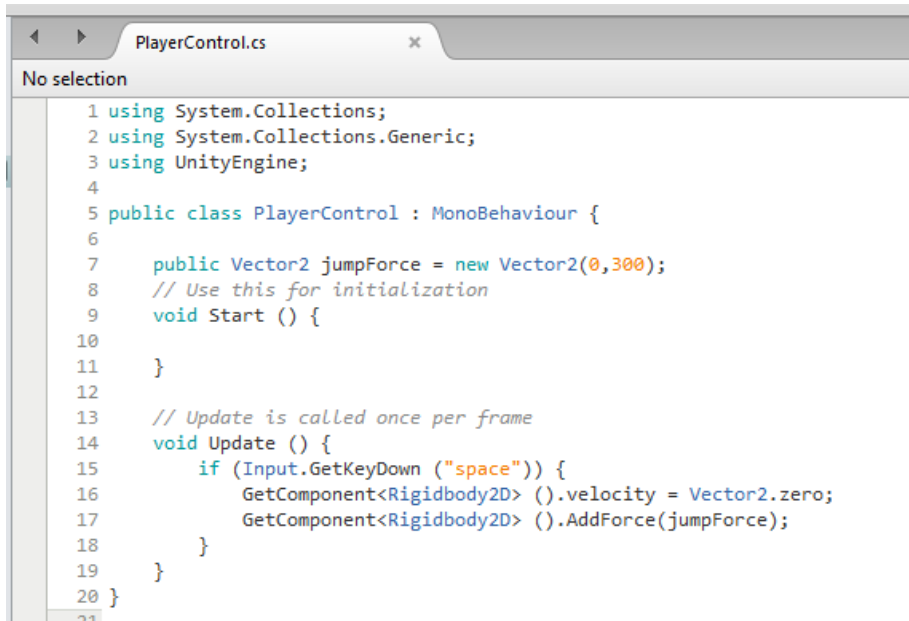
Game Player

Masukkan gambar player.png pada project windows kedalam scene view. Pada object player lakukan sesuatu berikut ini:

1. Rubah Inspector Transform posisi x dan y menjadi -7 dan 2, seperti pada gambar berikut:



2. Rubah property Inspector sprite renderer order in layer bernilai 1.
3. Tambahkan komponen Rigidbody2D pada Inspector object player.
4. Buat script baru pada object Player dengan nama “PlayerControl”, dengan rincian kode seperti berikut:



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerControl : MonoBehaviour {
6
7     public Vector2 jumpForce = new Vector2(0,300);
8     // Use this for initialization
9     void Start () {
10
11     }
12
13     // Update is called once per frame
14     void Update () {
15         if (Input.GetKeyDown ("space")) {
16             GetComponent<Rigidbody2D> ().velocity = Vector2.zero;
17             GetComponent<Rigidbody2D> ().AddForce(jumpForce);
18         }
19     }
20 }
21
```

JumpForce: merupakan Vector2 yang digunakan sebagai nilai ambang batas fungsi AddForce. AddForce: merupakan method yang hanya bisa diakses jika object memiliki komponen Rigidbody 2D didalamnya. Method ini memaksakan suatu pergerakan tiba-tiba yang diatur dengan nilai vector2. Dalam praktikum ini, karena efek gravitasi dari rigidbody, object akan selalu turun kebawah. Maka dari itu, kita akan memaksa player untuk seakan-akan melompat keatas saat spasi ditekan. Maka dari itu, yang diberi nilai hanya Vector 2 sumbu Y saja, yaitu dengan nilai 300. Velocity pada rigidbody 2D dibuat Vector2.Zero atau [0,0] untuk mengembalikan kecepatan objek menjadi 0.

5. Jalankan unity dan lihat apa yang terjadi jika keyboard spasi di tekan.
6. Tambahkan polygon collider 2D pada object player.
7. Rubah kode program PlayerControl menjadi seperti dibawah ini:

Tambahkan header library UnityEngine.SceneManagement:

```
using UnityEngine.SceneManagement;
```

digunakan untuk mengakses fungsi SceneManager.LoadScene(<nama scene>), dimana fungsi tersebut berguna untuk mereset player jika player melebihi tinggi scene atau jatuh ketanah atau menabrak rintangan-rintangan (obstacles).

Didalam update setelah scope dari Input.GetType(..) atau setelah kurung kurawal tutup dari scope Input.GetType(..), tambahkan:

```
Vector2 screenPosition =
Camera.main.WorldToScreenPoint (transform.position);
```

Kode program diatas digunakan untuk mengambil posisi x dan y dari player yang relatif terhadap world scene.

Setelah itu, tambahkan kode untuk memberi batasan lebih tinggi dan terjatuh:

```

if (screenPosition.y > Screen.height || screenPosition.y < 0)
    Die ();
kemudian, buat method Die() dibawah dari method Update():
void Die()
{
    Debug.Log ("game over");
    SceneManager.LoadScene ("Main");
}

```

Terakhir adalah tambahkan fungsi OnCollisionEnter2D, jika objek player nanti menabrak apapun rintangan atau obstacle akan mati.

```

void OnCollisionEnter2D(Collision2D coll)
{
    Die ();
}

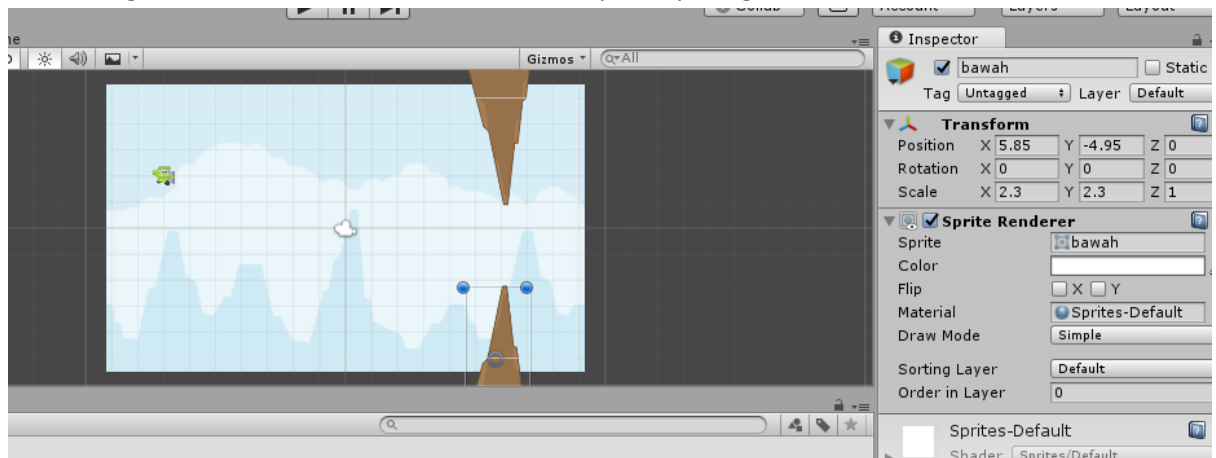
```

8. Jalankan unity dan lihat apa yang terjadi.

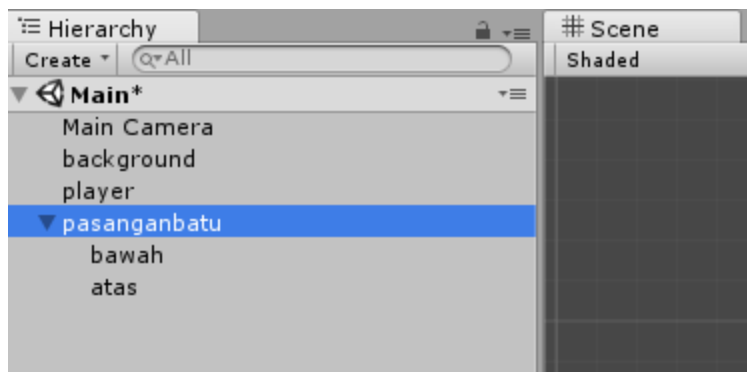
Game Obstacles

Selanjutnya adalah membuat rintangan-rintangan atau obstacles dari game flappycopter ini. Berikut adalah tahapannya:

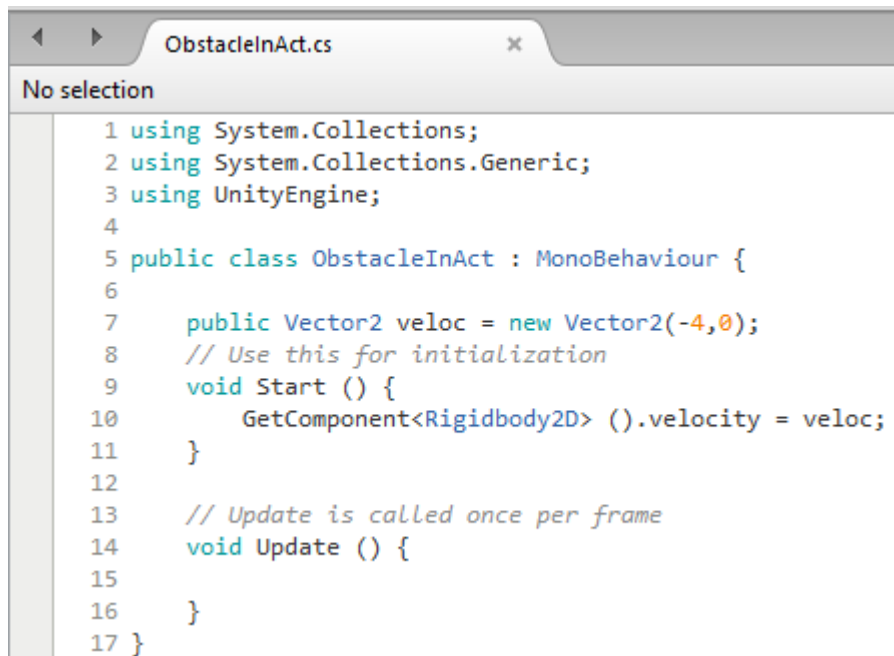
1. Masukkan gambar atas dan bawah kedalam Scene view. Pastikan order in layer bernilai 1.
2. Skala gambar atas dan bawah untuk x dan y bernilai 2.3.
3. Posisikan gambar atas dan bawah secara berhadapan seperti gambar berikut:



4. Tambahkan polygon collider 2D pada object atas dan bawah.
5. Buat game object kosong dengan cara klik GameObject->Create Empty di menu tab. Atau ctrl+shift+n.
6. Lihat hierarchy window dan rename "GameObject" menjadi "pasanganbatu".
7. Tarik object "atas" dan "bawah" kedalam "pasanganbatu" pada hierarchy window, seperti gambar berikut:



8. Tambahkan Rigidbody 2D pada Inspector dari game object "pasanganbatu" Kemudian rubah body type menjadi "Kinematic" dan jangan centang (uncheck) simulated atau pada versi unity 5.6 kebawah centang "isKinematic". Digunakan agar tidak terpengaruh oleh gravitasi dan pantulan yang diakibatkan benturan objek.
9. Buat script "ObstacleInAct" pada Inspector "pasanganbatu", seperti pada gambar berikut:



Kode tersebut berguna untuk menggerakkan pasangan batu secara mundur dengan kecepatan -4, ketika game berjalan.

10. Jalankan unity dan lihat apa yang terjadi.
11. Tarik object pasanganbatu dari hierarchy window kedalam folder Prefabs untuk agar dapat menjadi object prefab.
12. Hapus game object pasanganbatu dari hierarchy window.
13. Buat game objek kosong dan rename menjadi "ObstacleGenerator".
14. Tambahkan script baru pada Inspector "ObstacleGenerator" dengan nama "GenerateObs". Berikut adalah isi dari kode "GenerateObs":

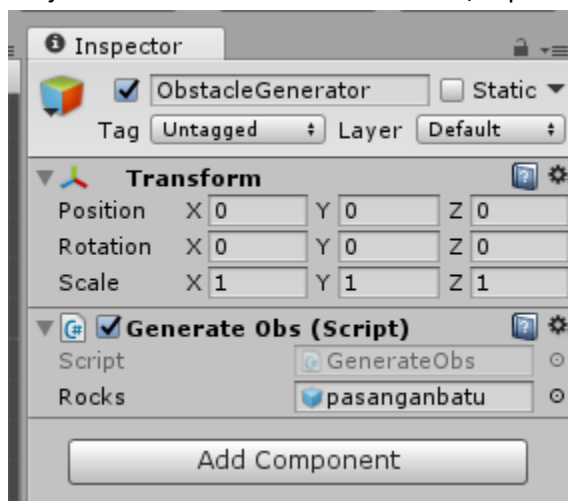
```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class GenerateObs : MonoBehaviour {
6
7     public GameObject rocks;
8     // Use this for initialization
9     void Start () {
10         InvokeRepeating ("CreateObstacle", 1f, 1.5f);
11     }
12     void CreateObstacle()
13     {
14         Instantiate (rocks);
15     }
16     // Update is called once per frame
17     void Update () {
18
19     }
20 }

```

Method/fungsi InvokeRepeating digunakan untuk memanggil method CreateObstacle setiap 1.5 detik dengan lama eksekusi dari method tersebut terbatas hanya 1 detik. Instantiate digunakan untuk instance object secara otomatis, disini akan dipanggil objek pasanganbatu.

15. Lihat pada Inspector dari ObstacleGenerator dan Tarik pasanganbatu dari folder Prefabs di Project Window kedalam variabel rock, seperti gambar dibawah ini:



16. Jalankan unity, dan lihat apa yang terjadi.
17. Agar pemain tidak bosan karena lokasi dari obstacle statis. Jadi buat lokasi obstacle menjadi random dengan cara menambahkan kode berikut dibawah GetComponent<Rigidbody2D>().velocity = veloc; pada script ObstacleInAct:

```

float range = 2;
transform.position = new Vector3 (transform.position.x,
    transform.position.y - range * Random.value,
    transform.position.z);

```

18. Jalankan Unity dan lihat apa yang terjadi.
19. Jika terjadi bug, batu terlalu kebawah maka perkecil nilai range.

Scoring System

Score akan menggunakan GUI Label pada Unity. Penjelasan GUI akan dijelaskan lebih lanjut pada pertemuan berikutnya. Namun demikian, kita akan mencoba membuat GUI Label untuk menampilkan score pada game flappycopter dengan dipanggil lewat kode. Berikut adalah tahapannya:

1. Masuk pada script "GenerateObs".
2. Buat variabel bertipe int dengan nama score dengan di-assign nilai 0.
3. Tambahkan Method OnGUI untuk menambahkan GUI Label pada scene game:

```
void OnGUI()  
{  
    GUI.color = Color.black;  
    GUILayout.Label ("Score: " + score.ToString());  
}
```

4. Tambahkan kode: score++; setelah Instantiate pada method CreateObstacle.
5. Jalankan Unity dan Voila!! Flappycopter sudah jadi!

Latihan (Optional)

Kembangkan game flappycopter dengan semua metode yang sudah diajarkan sebelumnya. Contohnya, kita dapat menambahkan koin atau life point. Gunakan kreatifitasmu!

KUMPULKAN:

Screenshoot game tab dan simpan dengan nama sc_04_01.jpg

REFERENSI

<http://zikky.lecturer.pens.ac.id/Tutorial%20Membuat%20Permainan%20Flappy%20Bird%20Menggunakan%20Unity3D.pdf>

Mekanisme Pengumpulan Praktikum Mingguan

Tidak ada yang harus dikumpulkan pada praktikum kali ini.