

COURSE OUTLINE

GAME PROGRAMMING

Oleh: Abas Setiawan

Contents (DAFTAR ISI)

Pendahuluan	1
Save & Load dengan PlayerPrefs	1
Highscore	2
Multi Level Game	4
Latihan (Opsional)	7
REFERENSI	7
Mekanisme Pengumpulan Praktikum Mingguan	8

Pendahuluan

Saat ini, kita sudah sampai pada akhir penghujung pembuatan Game 2D. Tidak akan ada lagi project 2D baru melainkan menambahkan fitur project game 2D yang sudah kita buat sebelumnya. Fitur yang belum ada itu adalah tentang mekanisme menyimpan dan membuka data yang ada pada game. Data tersebut dapat berupa score atau level game yang belum terbuka atau koleksi item.

Save & Load dengan PlayerPrefs

Penyimpanan data merupakan salah satu hal yang penting didalam pengembangan Game. Data yang disimpan dapat sangat berdampak terhadap permainan secara langsung. Fitur ini memungkinkan player untuk dapat melanjutkan permainan tanpa harus bermain dari awal lagi. Sehingga diperlukan mekanisme yang aman dan nyaman untuk player dapat menyimpan datanya. Metode konvensional untuk penyimpanan data biasanya menggunakan data serial yang di enkripsi (merubah dari data mentah menjadi data yang tidak bisa dibaca) dan disimpan ditempat yang tidak mudah dijangkau oleh player. Hal ini diperlukan karena bisa jadi player akan bertindak curang dengan merubah data sesuai yang dikehendaki.

Sayangnya mekanisme penyimpanan data secara serial tidaklah mudah. Selain itu mekanisme enkripsi dan deskripsi juga membutuhkan penanganan yang tepat serta rumit implementasinya. Unity sudah menyediakan mekanisme menyimpan data dan juga sudah di enkripsi secara otomatis dengan menggunakan PlayerPrefs. Cara penggunaan PlayerPrefs juga terbilang sangat mudah sekali, yaitu hanya dengan satu atau dua baris kode saja. Intinya adalah kita mengerti dahulu bagaimana cara menyimpan dan membuka data dengan mekanisme playerpref ini. Terlepas dari aman atau tidaknya playerpref jika dibandingkan dengan mekanisme enkripsi-deskripsi yang lain.

Mekanisme playerpref diawali dengan memanggil Class PlayerPrefs. Terdapat 3 jenis static method yang dapat digunakan untuk menyimpan data yaitu SetInt, SetFloat, dan SetString. Ketiga method tersebut memiliki dua parameter. Parameter pertama digunakan untuk memanggil nama kunci yang

didefinisikan sendiri oleh programmer. Parameter kedua merupakan nilai yang berkaitan dengan tipe data integer, float, atau string tergantung dari pemanggilan jenis methodnya.

```
PlayerPrefs.SetInt("<kunci>", nilai integer);
```

```
PlayerPrefs.SetFloat("<kunci>", nilai float);
```

```
PlayerPrefs.SetString("<kunci>", nilai string);
```

Di Windows, PlayerPrefs akan menyimpan kunci beserta nilainya di HKCU\Software\[company name]\[product name], dimana company name dan product name sudah diatur didalam project setting unity.

Di Windows Store, PlayerPrefs akan menyimpan kunci beserta nilainya di %userprofile%\AppData\Local\Packages\[ProductPackageId]>\LocalState\playerprefs.dat.

Di windows phone, PlayerPrefs akan menyimpan kunci beserta nilainya di application's local folder.

Di Android, PlayerPrefs akan menyimpan kunci beserta nilainya di SharedPreferences. SharedPreferences secara fisik dapat dilihat di direktori /data/data/pkg-name/shared_prefs/pkg-name.xml.

Di MacOS, PlayerPrefs akan menyimpan kunci beserta nilainya di /Library/Preferences folder, didalam file dengan nama, [company name].[product name].plist.

Untuk mengakses kunci atau membuka kunci atau load data, dapat dilakukan dengan memanggil tiga method berikut:

```
int x = PlayerPrefs.GetInt("<kunci>");
```

```
float x = PlayerPrefs.GetFloat("<kunci>");
```

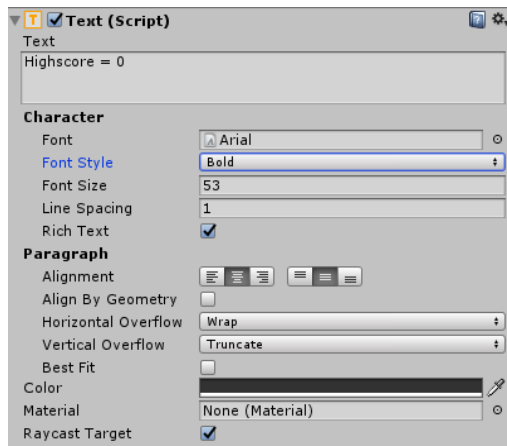
```
string x = PlayerPrefs.GetString("<kunci>");
```

Highscore

PlayerPrefs dapat dimanfaatkan untuk menyimpan score dari sebuah game. Dengan mekanisme menyimpan score pada suatu kunci di permainan pertama. Kemudian kunci tersebut dipanggil pada permainan kedua dan disana nilai kunci tersebut akan diperbaharui dengan syarat jika nilai kunci sebelumnya lebih rendah dibanding nilai score sekarang. Sehingga kita akan mendapatkan nilai score tertinggi pada semua permainan yang telah dilakukan.

PRAKTIKUM:

1. Buka project "Flappycopter" yang pernah dibuat sebelumnya dan buka scene menu.
2. Tambahkan UI text, rename dengan nama "TextHighscore".
3. Letakkan pada posisi x = 0 dan y = 200 sehingga berada pada tengah bagian bawah.
4. Lihat Inspector TextHighscore, pada property Text isikan seperti gambar berikut:



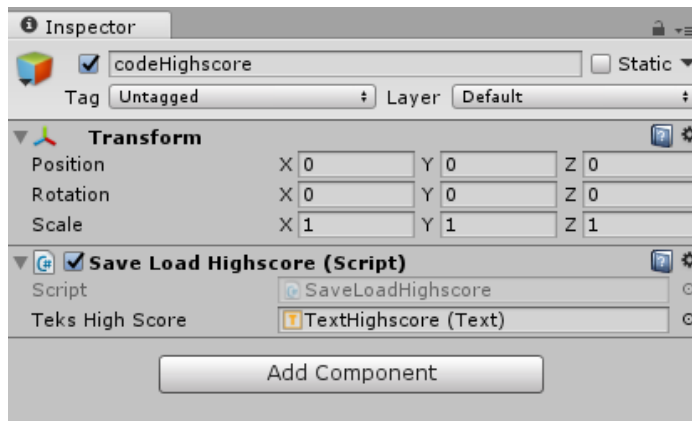
5. Buat game object kosong dan rename menjadi “codeHighscore”.
6. Tambahkan kode program “SaveLoadHighscore”:

```
SaveLoadHighscore ▶ SaveHighScore (int score)
1 using UnityEngine;
2 using UnityEngine.UI;
3
4 public class SaveLoadHighscore : MonoBehaviour {
5
6     public Text teksHighScore;
7     // Use this for initialization
8     void Start () {
9         teksHighScore.text = "Highscore = " + LoadHighScore().ToString();
10    }
11
12    public static int LoadHighScore()
13    {
14        int hg=0;
15        if (!PlayerPrefs.HasKey ("highscore"))
16            PlayerPrefs.SetInt ("highscore", 0);
17        else
18            hg = PlayerPrefs.GetInt ("highscore");
19        return hg;
20    }
21
22    public static void SaveHighScore(int score)
23    {
24        int hg = 0;
25        if (!PlayerPrefs.HasKey ("highscore"))
26            PlayerPrefs.SetInt ("highscore", 0);
27        else {
28            hg = PlayerPrefs.GetInt ("highscore");
29            hg += score;
30            PlayerPrefs.SetInt ("highscore", hg);
31        }
32    }
33 }
34
```

Terdapat 2 method static yaitu LoadHighScore() dan SaveHighScore, perhatikan bahwa method static disimpan pada memory common yang tidak membutuhkan instance dahulu ketika pemanggilan method tersebut.

PlayerPrefs.HasKey digunakan untuk mengecek apakah kunci tersebut sudah ada di local direktori atau belum.

7. Isikan TextHighScore pada Inspector codeHighscore seperti gambar berikut:



8. Buka kode GenerateObs, tambahkan kode berikut didalam method CreateObstacle setelah score++;
`SaveLoadHighscore.SaveHighScore (score);`
9. Jalankan dan mainkan game.
10. Lihat pada main menu setelah permainan ke-2.

KUMPULKAN:

Screenshoot game tab pada scene menu. Simpan dengan nama sc_08_01.jpg

Multi Level Game

Salah satu fitur yang dapat membuat player merasa tertantang saat memainkan game adalah level dari game tersebut. Level game biasanya dirancang dari tingkat yang termudah sampai tersulit. Perubahan setiap level game bermacam-macam, tergantung dari bagaimana developer merancang game mekanik yang ada. Beberapa perubahan yang diakibatkan oleh meningkatnya level diantaranya adalah:

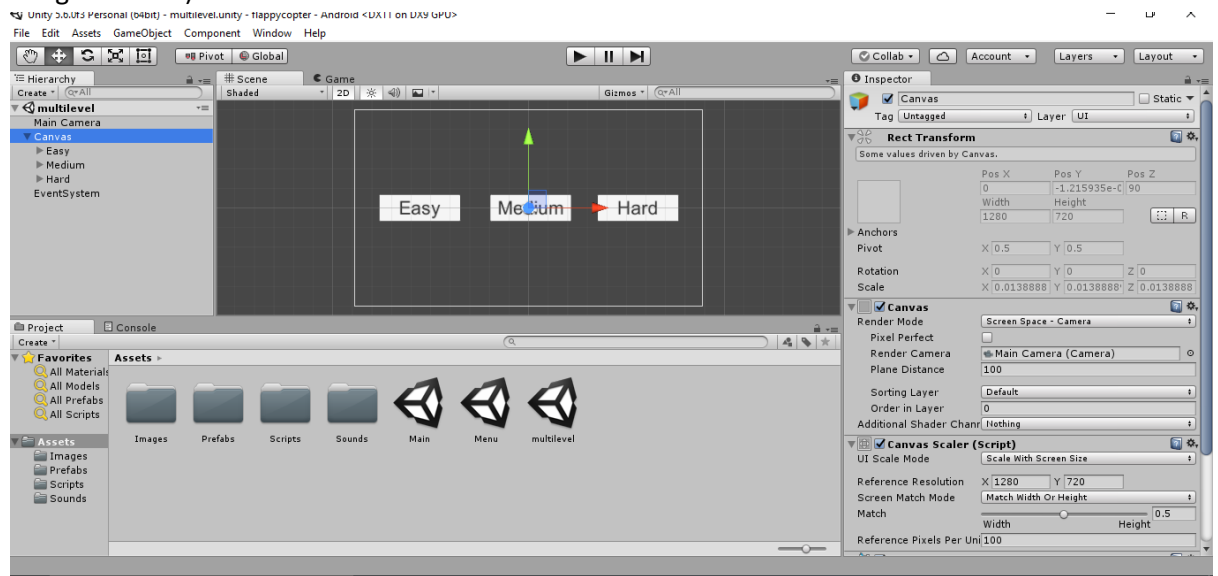
- membuat musuh lebih agresif ketika tingkat level bertambah,
- membuat kontrol player menjadi terbatas,
- merubah lingkungan/environment game,
- merubah game play

Namun demikian, kita tidak akan berfokus pada perubahan yang diakibatkan dari meningkatkan level. Kita akan berfokus pada bagaimana level baru yang tingkatannya lebih tinggi akan terbuka. Game developer biasanya menggunakan syarat supaya player dapat membuka level yang lebih tinggi. Syarat yang paling sederhana adalah dengan menyelesaikan level sebelumnya terlebih dahulu. Sebagai contoh, terdapat tiga level game yaitu easy, medium, hard. Untuk player yang bermain game pertama kalinya, diharuskan untuk menyelesaikan level easy terlebih dahulu. Player tidak dapat secara langsung membuka level medium atau hard. Pada praktikum ini, kita akan mencoba mengimplementasikan hal tersebut dengan memanfaatkan PlayerPrefs.

PRAKTIKUM:

1. Buka project "Flappycopter" yang pernah dibuat sebelumnya dan buat scene baru dengan nama "multilevel".

- Tambahkan 3 UI button dengan nama “Easy”, “Medium”, dan “Hard” (pastikan canvas diatur dengan benar!).



- Buka scene “menu”, copy object “codeGUI”, buka scene “multilevel”, paste object “codeGUI” pada hierarchy window.
- Buka script GUIManager.
- Tambahkan variabel bEasy,bMed,bHard bertipe Button. Pastikan sudah menambahkan using UnityEngine.UI dan using System.
- Tambahkan method static LoadLevel dan saveLevel:

```
public static int LoadLevel()
{
    int hg=0;
    if (!PlayerPrefs.HasKey ("level"))
        PlayerPrefs.SetInt ("level", 0);
    else
        hg = PlayerPrefs.GetInt ("level");
    return hg;
}

public static void saveLevel(int lvl)
{
    if (!PlayerPrefs.HasKey ("level"))
        PlayerPrefs.SetInt ("level", 0);
    else
        PlayerPrefs.SetInt ("level", lvl);
}
```

- Tambahkan method LoadButtonLevel untuk instance button dan mencari objek butto melalui nama objeknya(GameObject.Find).

```
void LoadButtonLevel()
{
    bEasy = GameObject.Find ("Easy").GetComponent<Button>();
    bMed = GameObject.Find ("Medium").GetComponent<Button>();
}
```

```

        bHard = GameObject.Find ("Hard").GetComponent<Button>
        bEasy.interactable = bMed.interactable =
        bHard.interactable = false;
    }

```

8. Rubah method Start() seperti berikut: (gunakan try catch untuk menghindari error null: <https://docs.unity3d.com/Manual/NullReferenceException.html>)

```

void Start () {
    try{
        LoadButtonLevel ();
        int levelstate = LoadLevel ();
        switch (levelstate) {
            case 0:
                bEasy.interactable = true;
                break;
            case 1:
                bEasy.interactable = true;
                bMed.interactable = true;
                break;
            case 2:
                bEasy.interactable = true;
                bMed.interactable = true;
                bHard.interactable = true;
                break;
        }
    }
    catch(NullReferenceException e) {
        //biarkan kosong atau boleh diberi keterangan null
    }
}

```

9. Edit method OnPlay yang menuju ke scene “multilevel” dan tambahkan method baru untuk runtime aksi dari setiap button ketika nanti button di klik. Sementara apapun button level yang di klik tetap akan ke scene Main.

```

public void OnPlay()
{
    SceneManager.LoadScene ("multilevel");
}

public void OnLevel1()
{
    SceneManager.LoadScene ("Main");
}

public void OnLevel2()
{
    SceneManager.LoadScene ("Main");
    //seharusnya buat scene baru untuk level medium
}

public void OnLevel3()
{
    SceneManager.LoadScene ("Main");
    //seharusnya buat scene baru untuk level hard
}

```

10. Isikan pada On Click Runtime Inspector Easy, Medium, dan Hard dengan GUIManager->OnLevel1(),GUIManager->OnLevel2(),dan GUIManager->OnLevel3().
11. Buka script PlayerControl dan edit method Die() dengan navigasi memanggil scene Menu, bukan memanggil scene Main. Diperlukan ketika player mati maka akan kembali ke menu game.
12. Buka script GenerateObs dan masuk pada method CreateObstacle. Buat aturan sederhana, jika obstacle terlihat atau score lebih dari sama dengan 2, maka level 2 terbuka, jika lebih dari sama dengan 4 maka level 3 terbuka.

```
void CreateObstacle()
{
    Instantiate (rocks);
    score++;
    SaveLoadHighscore.SaveHighScore (score);
    if (score >= 2)
        GUIManager.saveLevel (1);
    if(score >= 4)
        GUIManager.saveLevel (2);
}
```

13. Buka scene menu, jalankan unity, mainkan game, dan amati apa yang terjadi.

KUMPULKAN:

Screenshoot game tab pada scene multilevel saat pertama kali bermain. Simpan dengan nama sc_08_02.jpg

Screenshoot game tab pada scene multilevel setelah score sama dengan atau lebih dari 2. Simpan dengan nama sc_08_03.jpg

Screenshoot game tab pada scene multilevel setelah score sama dengan atau lebih dari 4. Simpan dengan nama sc_08_04.jpg

Latihan (Opsional)

Jika sudah selesai praktikum, silahkan kembangkan game yang ada pada praktikum kali ini dengan pengetahuan yang sebelumnya sudah dipelajari dan kreativitas kamu. Coba buat collect item dengan memanfaatkan playerprefs.

REFERENSI

<https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>

<https://unity3d.com/learn/tutorials/topics/scripting/high-score-playerprefs>

<http://answers.unity3d.com/questions/27973/idiots-guide-to-player-prefs.html>

<http://unitycsharp.blogspot.co.id/2016/03/unity-c-tutorial-indonesia-basic-28.html>

Mekanisme Pengumpulan Praktikum Mingguan

Untuk setiap mahasiswa: Buat 1 folder beri nama NIM lengkap. Masukkan file screenshot sc_08_01.jpg, sc_08_02.jpg, sc_08_03.jpg dan sc_08_04.jpg. Kemudian rar atau zip folder NIM kamu tadi dan kirimkan ke ketua kelas.

Contoh: A11200904997.rar

Untuk Ketua Kelas: Slapkan folder dengan nama Kode Kelas “_08” yang berisi kumpulan praktikum setiap mahasiswa. Lalu buat rar atau zip dan kirimkan ke email dosen.

Contoh: A114401_08.rar