MODUL 2
CORE UI

A. TUJUAN

- Mahasiswa dapat membuat tombol UI untuk pergerakan antar layar (scene)
- Mahasiswa dapat mengatur gambar didalam panel dan mengatur kedalaman gambar dengan menggunakan tombol
- Mahasiswa dapat membuat UI slider yang interaktif
- Mahasiswa dapat membuat penghitung waktu (countdown timer) dengan menggunakan UI slider
- Mahasiswa dapat membuat komponen input field untuk memasukkan teks (text entry)
- Mahasiswa dapat membuat toggle basic
- Mahasiswa dapat membuat radio buttons dengan meggunakan toggle group

B. PETUNJUK

1. Awali setiap kegiatan praktikum dengan berdoa
2. Baca dan pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik
3. Kerjakan tugas-tugas praktikum dengan baik, sabar dan jujur
4. Tanyakan kepada dosen apabila ada hal-hal yang kurang jelas

C. ALOKASI WAKTU : 3 jam pelajaran

D. DASAR TEORI

1. Button UI

Suatu game memiliki beberapa menu yang terbagi pada beberapa layar (scene), seperti menampilkan instruksi, high scores, level pemain, pengaturan suara, dan lain-lain. Unity menyediakan tombol UI untuk mempermudah user melakukan action di setiap layarnya seperti perpindahan layar satu ke layar yang lainnya.

2. Animation Button UI

Untuk menciptakan button yang interaktif, maka dapat menggunakan cara menambahkan animasi pada button ketika mouse-over. Cara yang paling sederhana adalah dengan memberikan warna yang berbeda pada button tersebut. Cara yang lain adalah membuat button yang memiliki highlight dinamik, yaitu saat cursor berada tepat diatas button, maka ukuran button akan terlihat lebih besar.

3. **Panel Depth**

   Sibling depth adalah suatu metode yang bertujuan agar objek UI akan tampil diatas atau dibawah dengan objek UI yang lain. Objek tersebut dapat berupa panel yang berisikan image. Terdapat bagian dari sibling depth pada unity, yaitu :

   a. SetAsLastSibling = metode yang digunakan untuk memindahkan posisi objek menjadi paling atas.
   b. SetAsFirstSibling = metode yang digunakan untuk memindahkan posisi objek menjadi paling bawah.
   c. MoveUpOne = metode yang digunakan untuk menaikkan posisi objek satu kali keatas objek lainnya.
   d. MoveDownOne = metode yang digunakan untuk menurunkan posisi objek satu kali kebawah objek lainnya

4. **Countdown Timer**

   Countdown timer adalah salah satu objek yang biasanya digunakan pada game untuk proses loading progress bar, penghitung waktu dan nyawa, perhitungan tenaga dari player, dan lain-lain.

5. **Toggles**

   Untuk memilih dua pilihan pada menu game dapat menggunakan toggles. Contoh toggle yang sering digunakan adalah pemilih sound on/off, Single/Multiple Player, dan lain-lain.

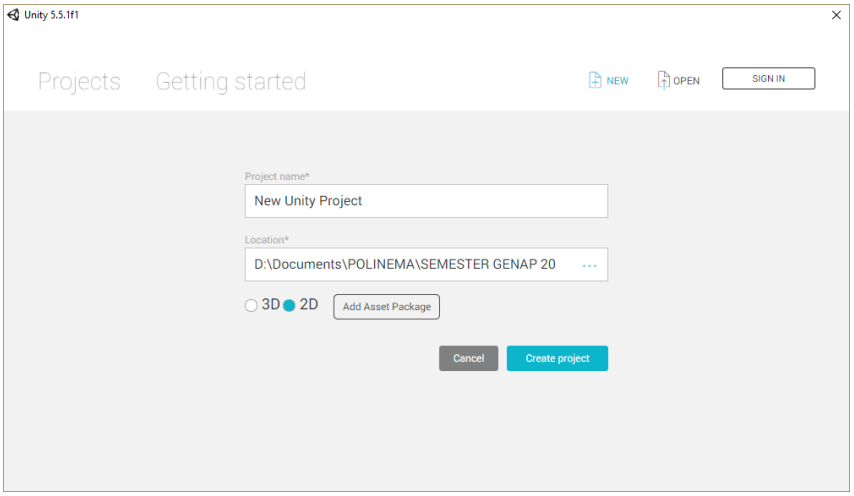
6. **Radio Button**

   Radio button berisikan kumpulan tombol-tombol yang digunakan untuk memilih salah satu dari beberapa pilihan. Misal pada game, digunakan untuk memilih level permainan (Easy/Medium/Hard). Radio button dapat dibuat dari toggle group dan ditambahkan beberapa logic untuk men-sinkronisasikan hubungan antar tombol.

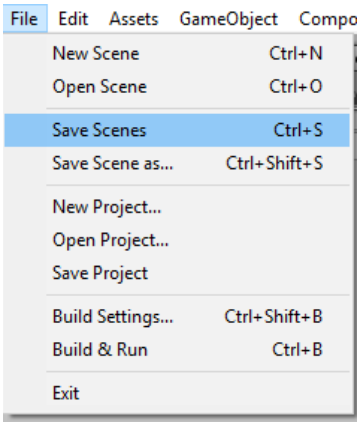
E. LATIHAN PRAKTIKUM

Membuat tombol UI untuk pergerakan antar layar (scene)

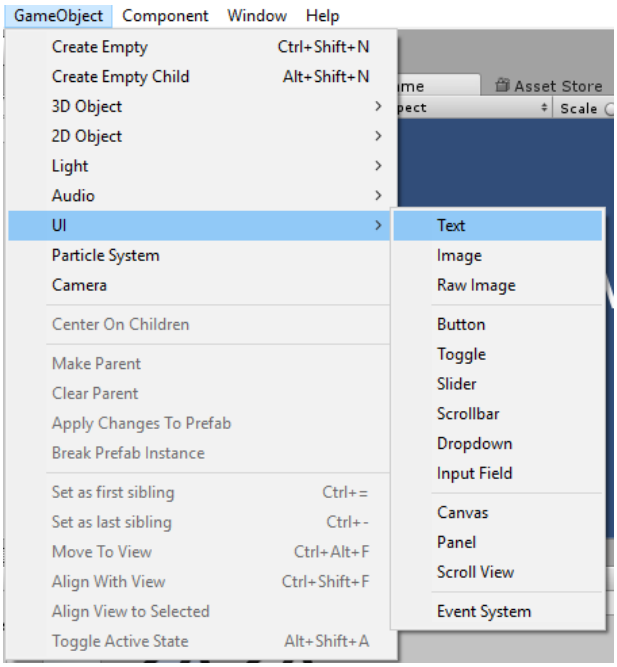
1. Buatlah projek baru 2D pada unity.

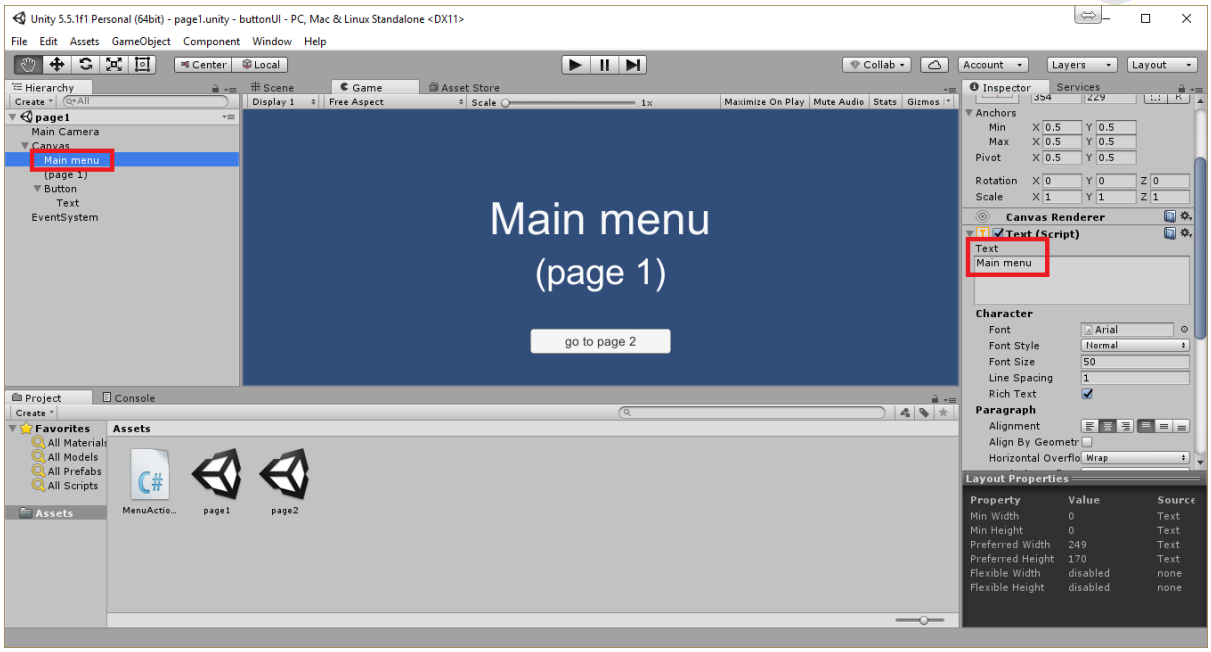


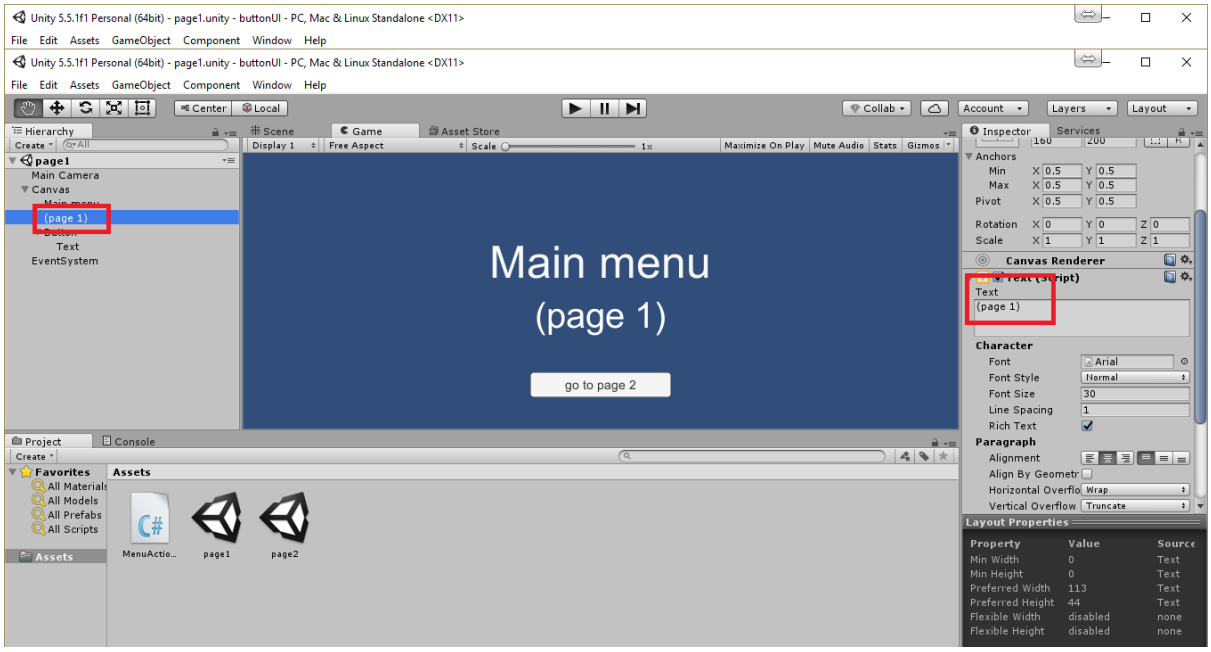
2. Save layar kosong tersebut dengan nama **page1**.



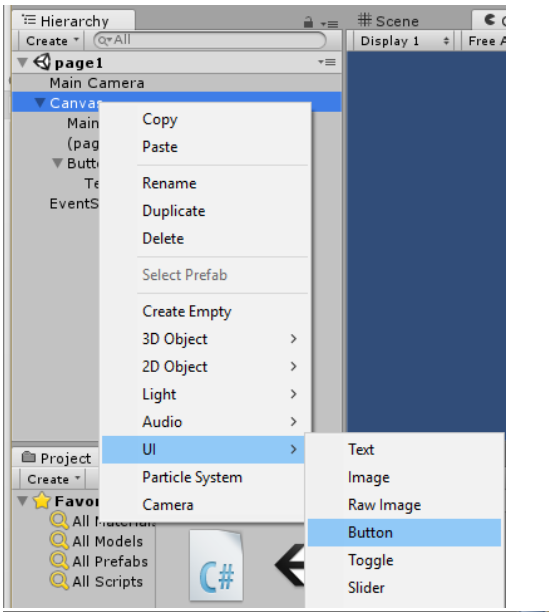
3. Tambahkan dua objek **UI Text** dimana posisinya adalah top center pada layar, dan isikan pula text berupa **`Main Menu`** dan **`(page 1)`** dengan ukuran huruf yang besar. Jangan lupa untuk mengubah nama text sesuai dengan isinya.

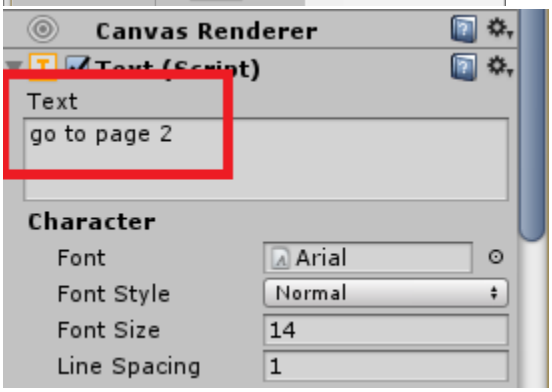




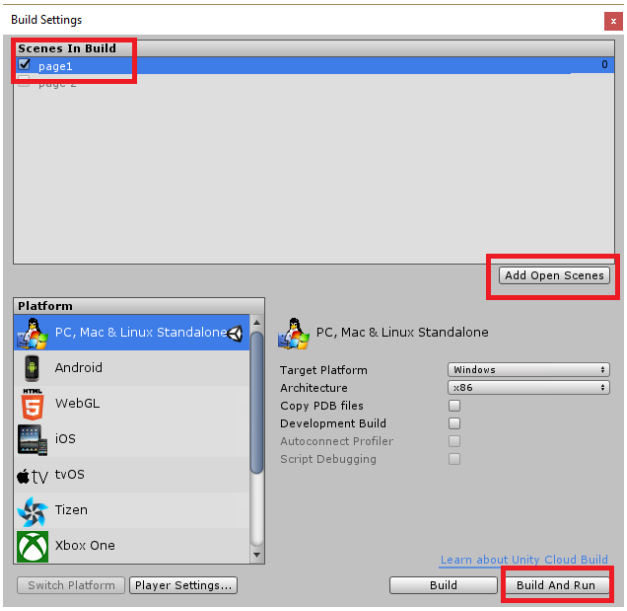


4. Tambahkan **Button UI** dengan posisi middle center pada layar. Caranya adalah pada hierarchy panel, klik kanan pada **Canvas → UI → Button**. Masukkan text pada button berupa tulisan **`go to page 2.`**

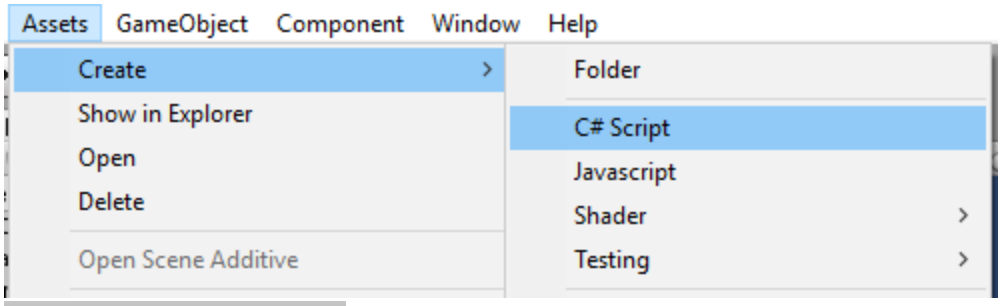


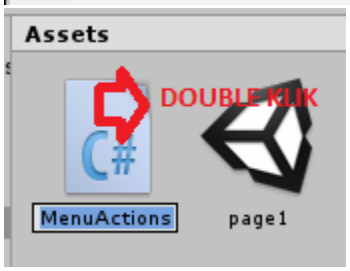


5. Build layar tersebut, dengan memilih menu **File → Build Settings**. Kemudian, klik button **Add Current,** sehingga layar page 1 menjadi layar pertama pada daftar Scene in the Build. Setelah selesai, klik button **Build & Run** untuk menjalankan layar.



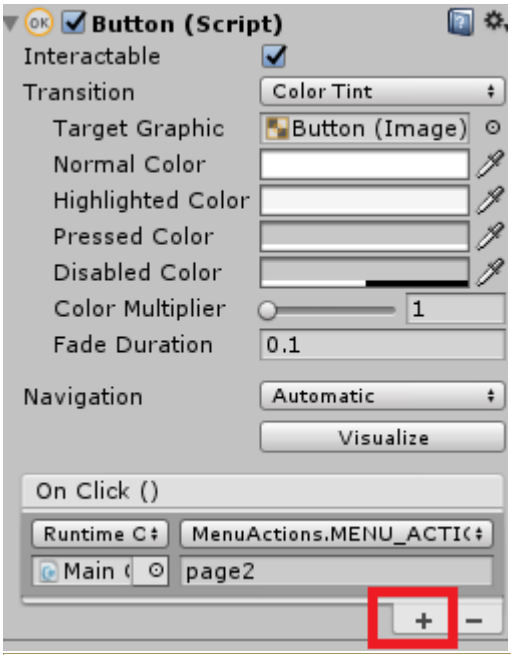
6. Buatlah script C# dengan nama **`MenuActions`**, dimana mengikuti source code dibawah ini. Script tersebut dimasukkan sebagai komponen kedalam **Main Camera**, dengan cara drag file script C# tersebut kedalam **Main Camera** pada **Hierarchy**.

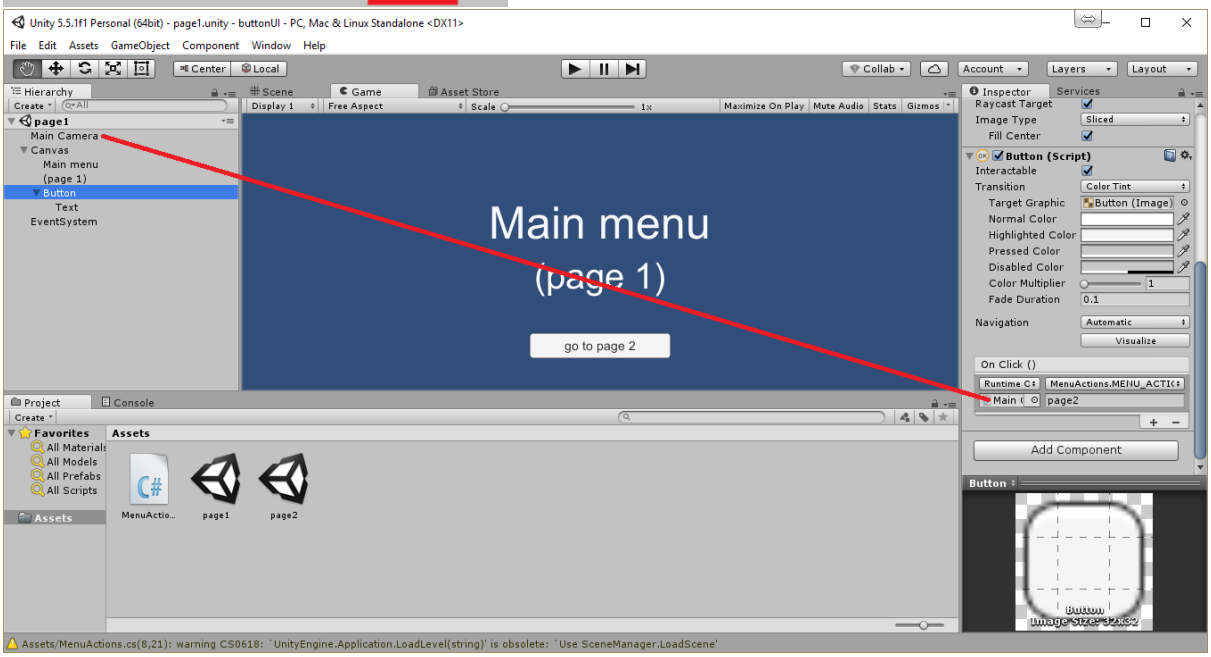




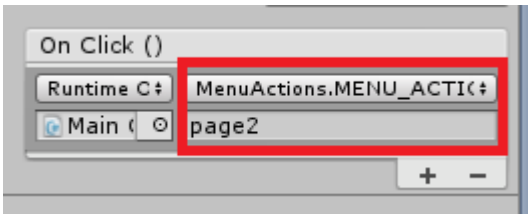

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MenuActions : MonoBehaviour {
    public void MENU_ACTION_GotoPage(string sceneName)
    {
        Application.LoadLevel(sceneName);
    }

}
```

7. Pastikan **Button** telah dipilih pada **Hierarchy** dan klik button plus "+" pada Inspector view untuk membuat **OnClick** Event yang baru pada button ini.
8. Drag **Main Camera** dari **Hierarchy** pada kolom dibawah **Runtime Only.** Ini berarti ketika **Button** menerima Event, maka akan memanggil sebuah method dari script yang berada di dalam **Main Camera.**

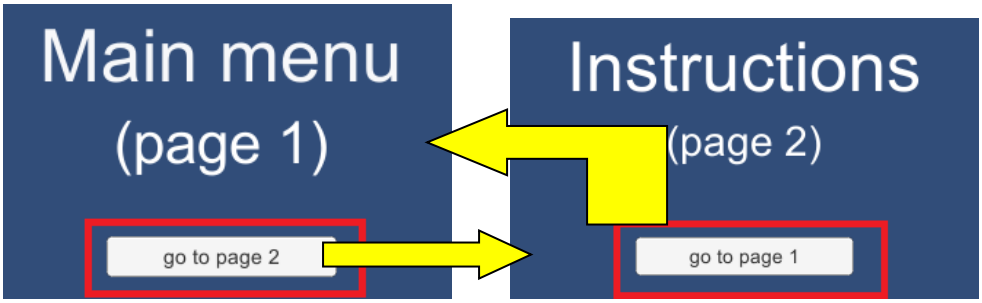




9. Pilih method **MENU_ACTION_GotoPage()** dari **MenuActions.** Kemudian untuk kolom yang berisikan text **No function,** ubah dengan mengetik **page 2,** dimana page 2 ini adalah fungsi yang akan di-passing-kan kedalam method ketika button menerima pesan dari **OnClick** event.

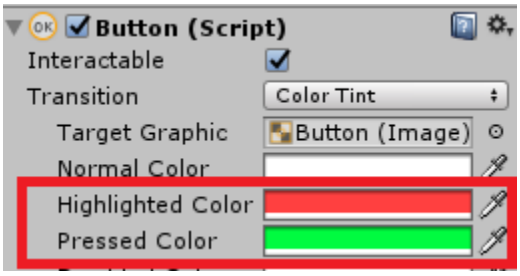


10. Save layar tersebut. Kemudian, buatlah layar kosong yang baru dan save dengan nama **page2.**
11. Ikuti langkah-langkah yang sama pada layar **page2.** Tambahkan GameObject **UI Text** dengan text berisikan **Instructions** dan **(page 2)** dengan ukuran huruf yang besar. Tambahkan **UI Button,** dimana berisikan text **go to page 1.**
12. Pada layar **page2**, drag script **MenuActions** kedalam **Main Camera.**
13. Pilih method **MENU_ACTION_GotoPage()** dari **MenuActions.** Kemudian untuk kolom yang berisikan text **No function,** ubah dengan mengetik **page 1,** dimana page 1 ini adalah fungsi yang akan di-passing-kan kedalam method ketika button menerima pesan dari **OnClick** event.
14. Save layar **page2.**
15. Tambahkan layar page2 kedalam daftar build (**File→Build Setting→ Add Open Scenes**), sehingga sekarang layar page1 dan page2 telah berada pada daftar build.
16. Run dengan klik tombol **Build and Run**, atau dengan cara tekan tombo pada keyboard (ctrl+B)
17. Hasil Output seperti berikut ini :

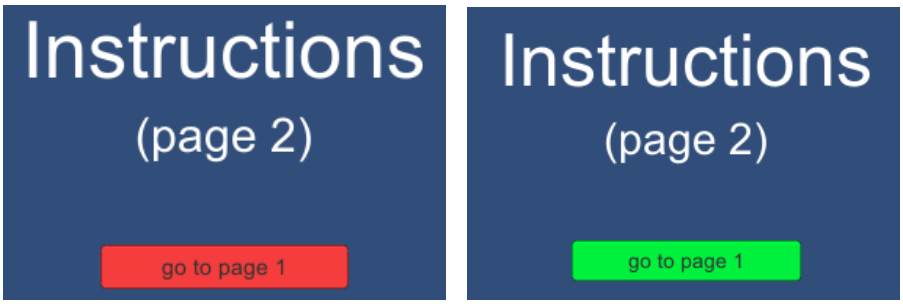


Pada page 1, jika ditekan tombol **go to page 2** akan menampilkan layar page 2. Sebaliknya, pada page 2 jika ditekan tombol **go to page 1** akan menampilkan layar page 1.

18. Untuk menambahkan animasi pada button ketika mouse-over, dapat mengubah warna pada button tersebut saat mouse-over **(Highlighted Color)** atau ditekan **(Pressed Color).**

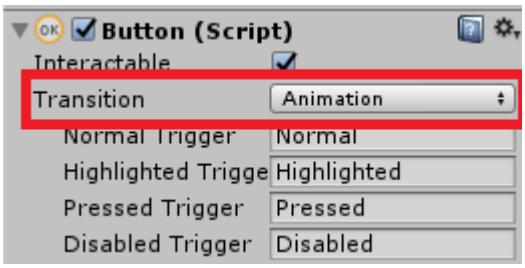


19. Hasilnya seperti berikut ini :

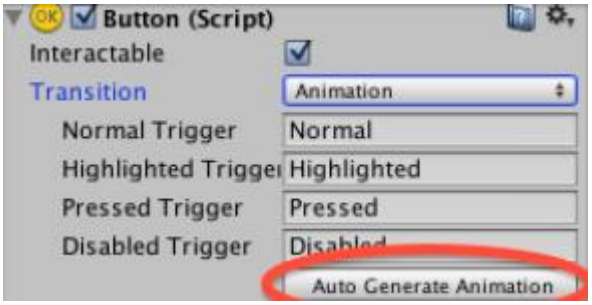


Membuat Animasi pada Button Mouse-Over dengan Highlight Dinamik

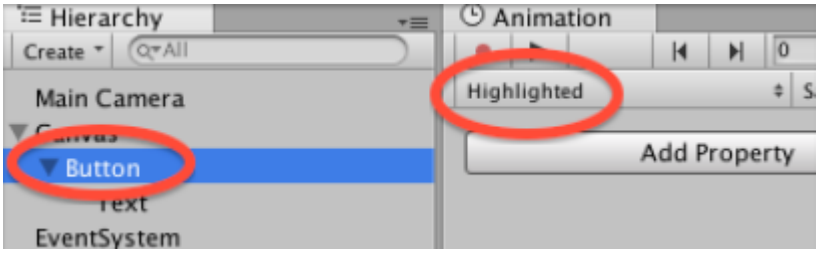
1. Buatlah 2D projek unity yang baru.
2. Buatlah sebuah **Button (GameObject → UI → Button).**
3. Pastikan **Button** sedang dipilih (diklik) pada **Hierarchy**. Pada komponen **Inspector Button (Script)**, atur property **Transition** menjadi **Animation.**



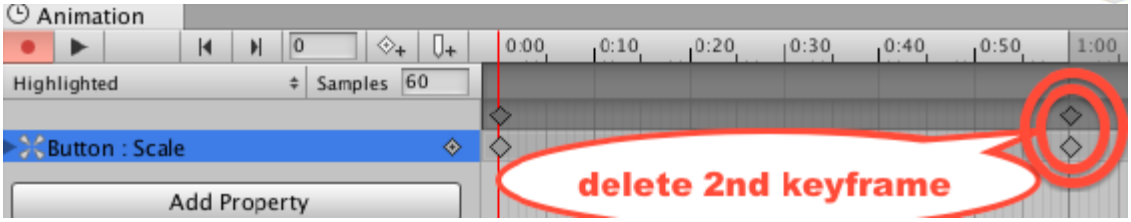
4. Klik button **Auto Generate Animation**



5. Save button dengan nama **button-animation-controller**
6. Pastikan **Button** dipilih pada **Hierarchy.** Kemudian, pada **Animation** panel,pilih **Highlighted.**



7. Pada **Animation** panel, klik tombol record merah dan kemudian klik tombol **Add Property.** Pilih **Rect Transform → Scale.**
8. Dua frame telah terbuat. Hapus frame yang kedua pada detik 1.00.

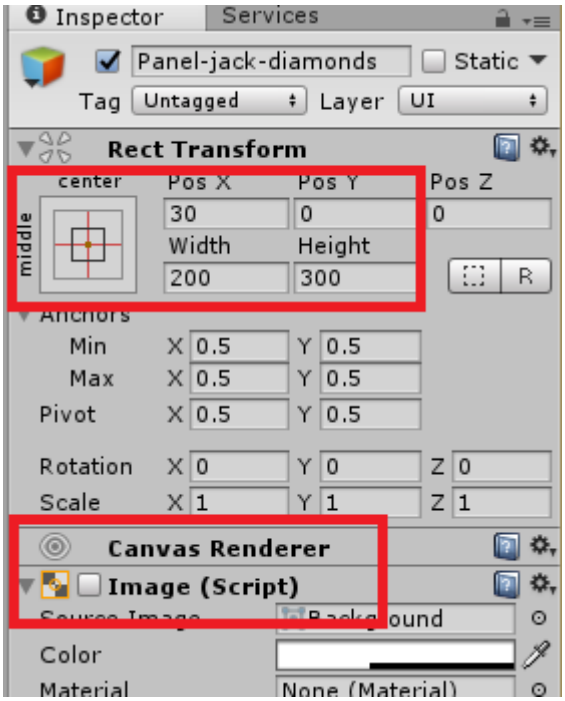


9. Pilih frame pertama pada detik ke 0.00. Kemudian pada **Inspector view**, set scale X dan Y pada **Rect Transform** menjadi (1.2, 1.2).
10. Klik tombol record merah pada pojok kiri atas untuk kedua kalinya agar dapat menghentikan proses rekaman perubahan dari animasi.
11. Save dan run (ctrl+b).
12. Hasil output :

Jika cursor mouse diarahkan pada button, maka ukuran button akan lebih besar.

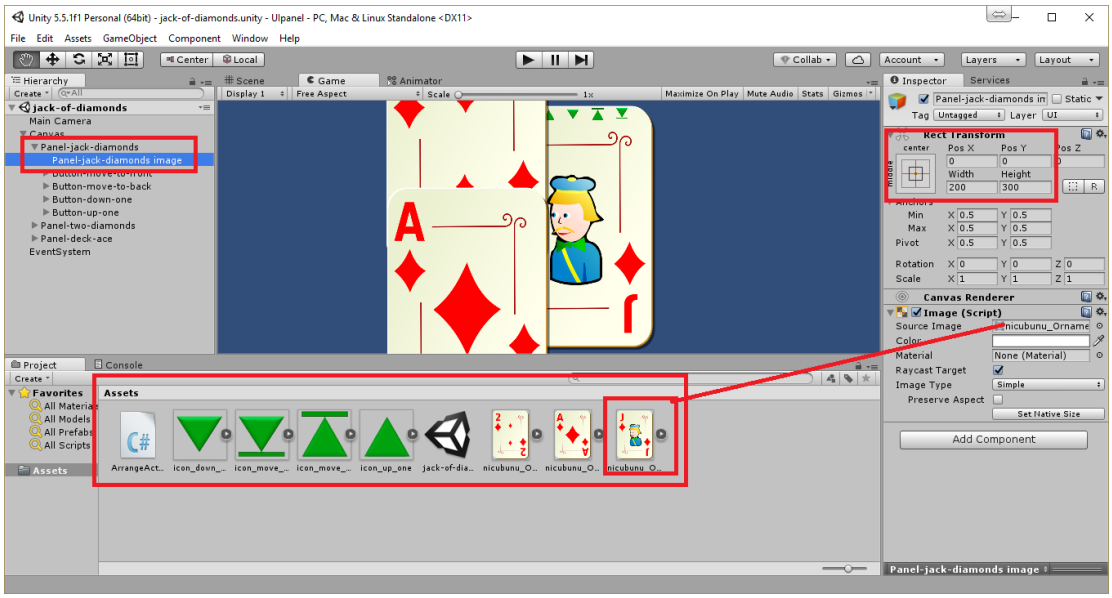
- **Mengatur gambar didalam panel dan mengatur kedalaman gambar dengan menggunakan tombol**

1. Untuk proses ini memerlukan **Asset** pada folder 1362_01_08.
2. Buatlah 2D projek unity yang baru.
3. Buatlah sebuah **UI Panel** yang baru dengan nama **Panel-jack-diamonds.** Posisikan pada middle-center dari layar dengan ukuran wide 200 piksel dan high 300 piksel. Uncheck komponen **Image (Script)** dari panel ini (tidak menampilkan kotak semi-transparan pada panel).

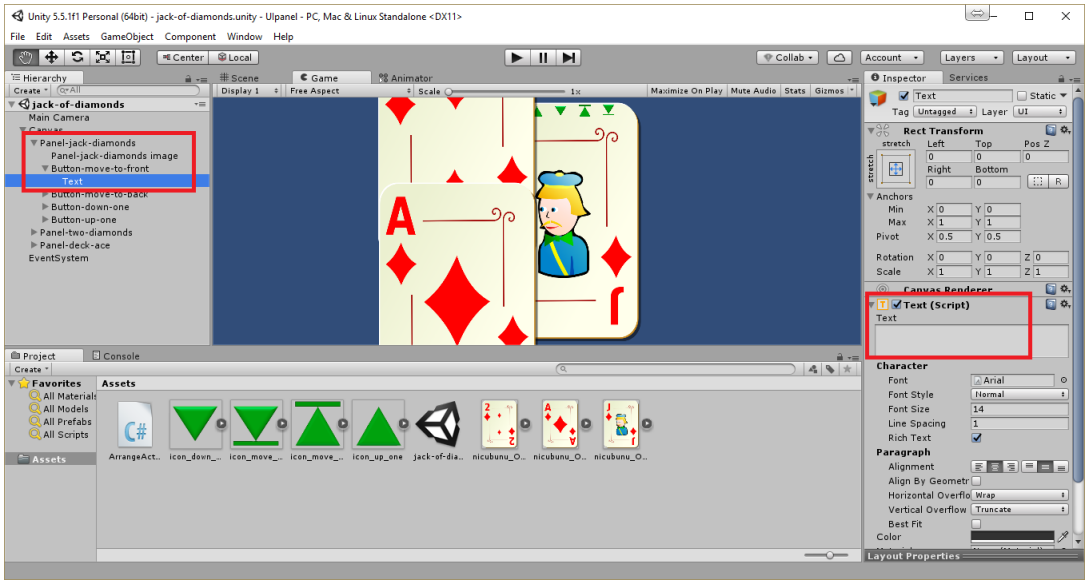


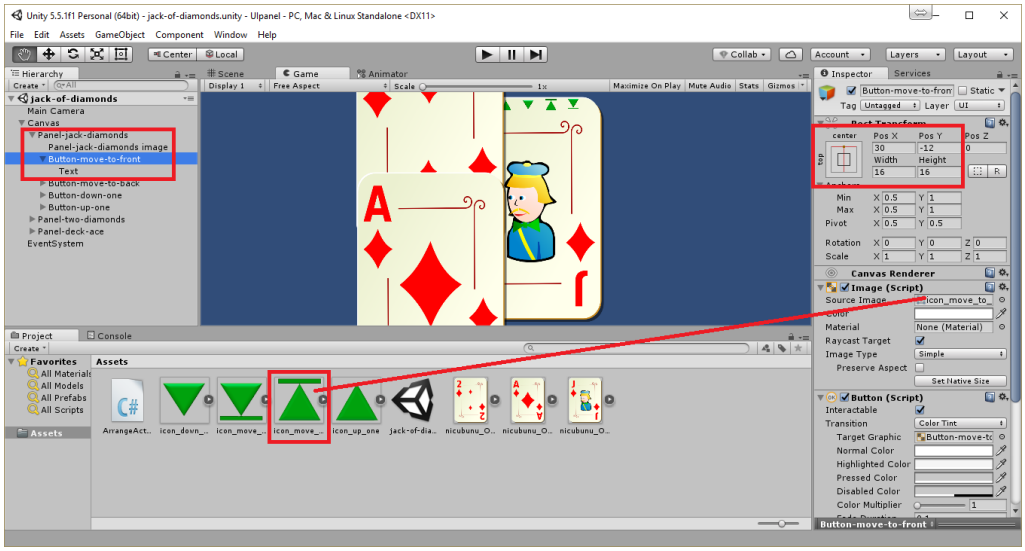
4. Buatlah sebuah **UI Image** dan jadikan image ini sebagai child dari **Panel-jack-diamonds.**
5. Posisikan **Panel-jack-diamonds image** pada center-middle dengan ukuran 200 x 300. Drag gambar **Jack-of-diamonds** (Asset) kedalam property **Source Image** pada

komponen **Image (Script)** di **Inspector tab.**

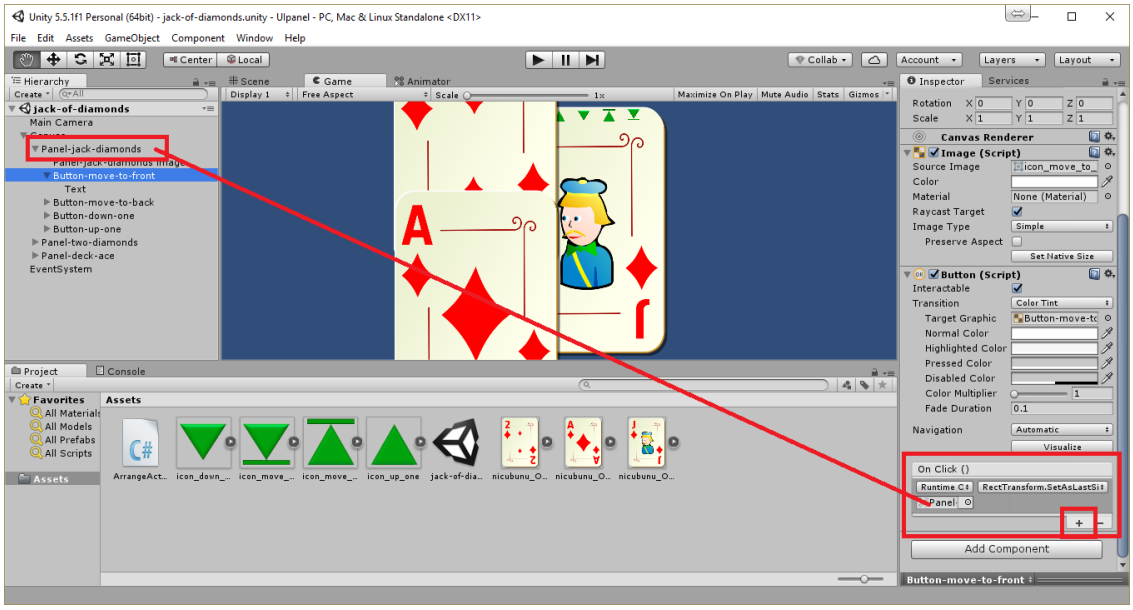


6. Buatlah **UI Button** dengan nama **Button-move-to-front.** Jadikan button ini sebagai child dari **Panel-jack-diamonds.** Hapus isi text dari button tersebut (hanya memerlukan button sebagai icon).
7. Ukuran dari **Button-move-to-front** adalah 16 x 16 dengan posisi top-center dari gambar kartu sehingga akan terlihat seperti pada bagian top dari gambar kartu. Drag gambar icon dengan nama **icon_move_to_front** (Asset) kedalam **Source Image,** untuk komponen **Image (Script)** di **Inspector view.**





8. Pastikan **Button-move-to-front** dipilih pada **Hierarchy**. Kemudian, klik tanda plus "+" pada bagian bawah dari komponen **Button (Script)** di **Inspector view** yaitu kolom **OnClick.**
9. Drag **Panel-jack-diamonds** dari **Hierarchy** kearah **Object slot.**
10. Untuk daftar method, pilih **RectTransform.SetASLastSibling.**

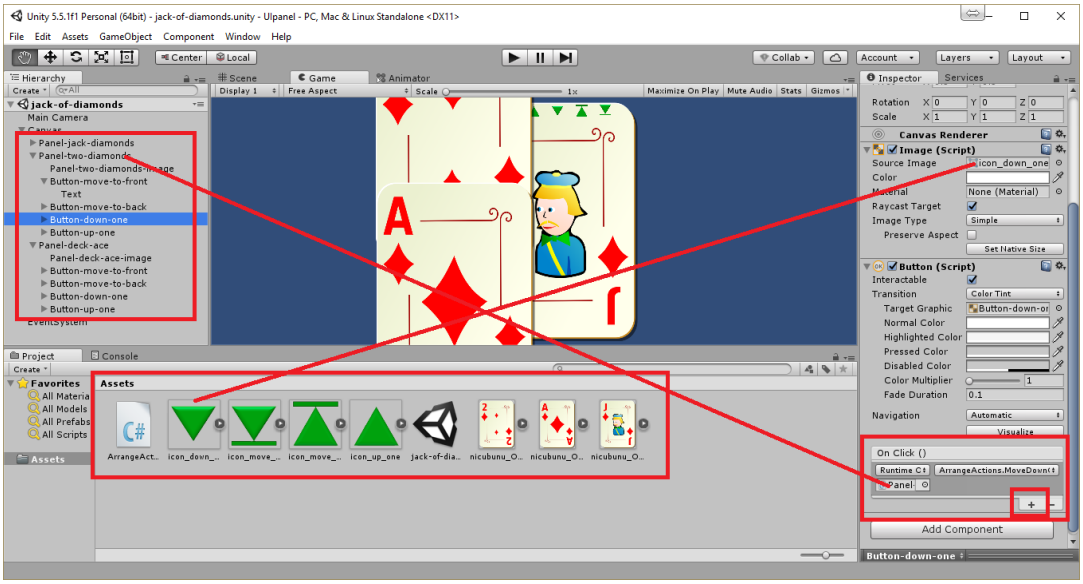


11. Ulangi step 3. Buatlah Panel kedua dengan nama **Panel-two-diamonds.** Posisikan sedikit kekanan dari arah **Panel-jack-diamonds.**
12. Ulangi step 3. Buatlah Panel ketiga dengan nama **Panel-ace-diamonds.** Posisikan sedikit kebawah dari arah **Panel-jack-diamonds.**
13. Buatlah script C# dengan nama **ArrangeActions,** dimana berisikan source code seperti dibawah ini. Kemudian masikkan script ini kepada setiap panel (3 panel) dengan cara

drag file script pada menuju ke setiap panel pada **Hierarchy.**

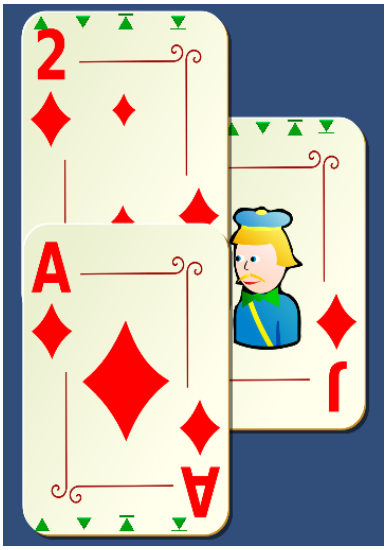
```
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.EventSystems;
using System.Collections;
public class ArrangeActions : MonoBehaviour
{
    private RectTransform panelRectTransform;

    void Start()
    {
        panelRectTransform = GetComponent<RectTransform>();
    }
    public void MoveDownOne()
    {
        print("(before change) " + gameObject.name + "sibling index = " + panelRectTransform.GetSiblingIndex());
        int currentSiblingIndex =
        panelRectTransform.GetSiblingIndex();
        panelRectTransform.SetSiblingIndex(currentSiblingIndex
        - 1);
        print("(after change) " + gameObject.name + " sibling index = " + panelRectTransform.GetSiblingIndex());
    }
    public void MoveUpOne()
    {
        print("(before change) " + gameObject.name + "sibling index = " + panelRectTransform.GetSiblingIndex());
        int currentSiblingIndex =
        panelRectTransform.GetSiblingIndex();
        panelRectTransform.SetSiblingIndex(currentSiblingIndex
        + 1);
        print("(after change) " + gameObject.name + " sibling index = " + panelRectTransform.GetSiblingIndex());
    }
}
```

14. Tambahkan button kedua untuk setiap panel. Gunakan icon dengan nama **icon_move_to_back** (Asset) dan atur event **OnClick** dengan fungsi **SetAsFirstSibling.**
15. Tambahkan dua button lainnya untuk setiap panel dengan nama **icon_down_one** (Asset) dengan memanggil fungsi **MoveDownOne()** dan **icon-up-one** (Asset) dengan memanggil fungsi **MoveUpOne().**
16. Aturlah posisi keempat button dari setiap panel tersebut agar tetap terlihat walaupun dalam keadaan saling bertumpukan



17. Save dan run layar unity (ctrl+b).

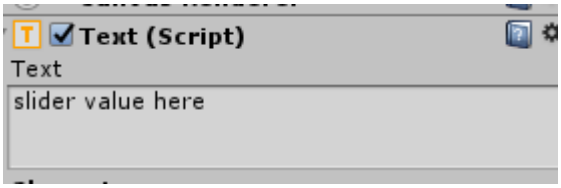
18. Hasil output :



Terdapar tiga kartu yang saling bertumpuk sesuai dengan hierarchy-nya. Jalankan keempat tombol dari setiap kartu untuk memindahkan kartu sesuai dengan keiinginan.

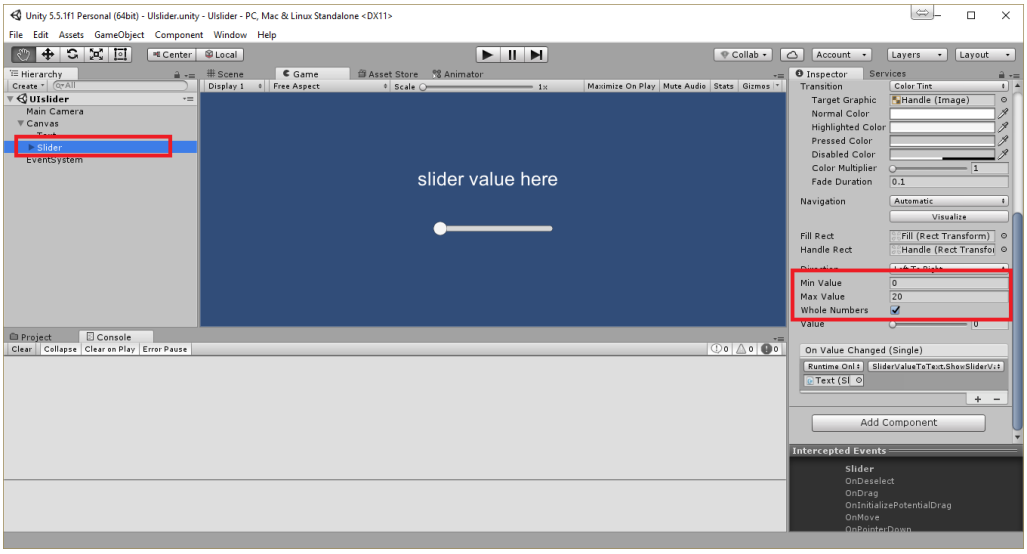
Membuat UI slider yang interaktif

1. Buatlah sebuah 2D projek unity yang baru.
2. Tambahkan **UI Text** (GameObject→UI→Text) dengan ukuran huruf 25. Edit isi teks menjadi **slider value here** (teks ini akan diganti dengan **slider value** ketika layar dijalankan).



3. Pada **Hierarchy**, tambahkan GameObject **UI→Slider.**
4. Pada tab **Inspector,** mdifikasikan pengaturan **Rect Transform** untuk posisi slider dengan top-middle dari layar.

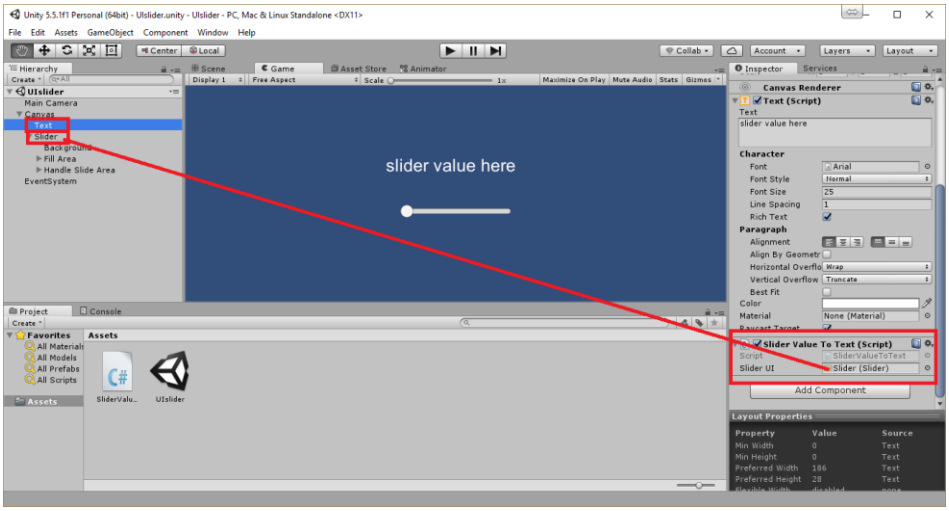
5. Aturlah **Min Value** dan **Max Value** menjadi 0 dan 20, kemudian centang checkbox **Whole Numbers.**



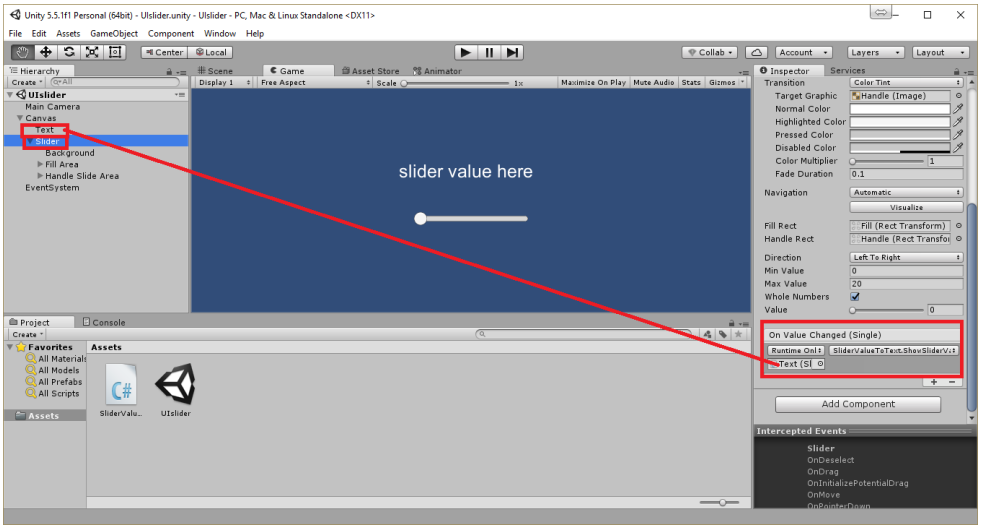
6. Buatlah script C# dengan nama **SliderValueToText,** dimana memiliki source code seperti dibawah ini. Kemudian, masukkan script tersebut kedalam komponen **Text** pada **Hierarchy.**

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class SliderValueToText : MonoBehaviour
{
    public Slider sliderUI;
    private Text textSliderValue;
    void Start()
    {
        textSliderValue = GetComponent<Text>();
        ShowSliderValue();
    }
    public void ShowSliderValue()
    {
        string sliderMessage = "Slider value = " +
        sliderUI.value;
        textSliderValue.text = sliderMessage;
    }
}
```

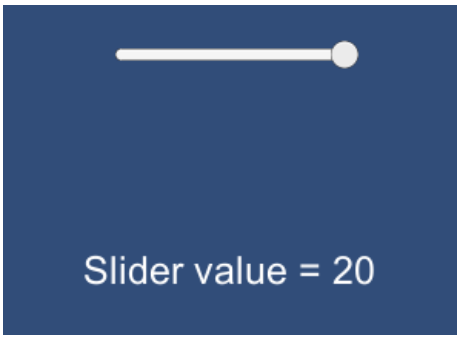
7. Pastikan GameObject **Text** dipilih pada **Hierarchy.** Kemudian, pada **Inspector view,** drag **Slider** kedalam variable **Slider UI** yang berada pada script **SliderValueToText.**



8. Pastikan GameObject **Slider** dipilih pada **Hierarchy.** Kemudian, pada **Inspector view,** drag GameObject **Text** kedalam **None(Object)** pada bagian **On Value Changed (Single).**
9. Pada bagian **On Value Changed (Single),** pilih method **SliderValueToText**, kemudian pilih fungsi **ShowSliderValue().**



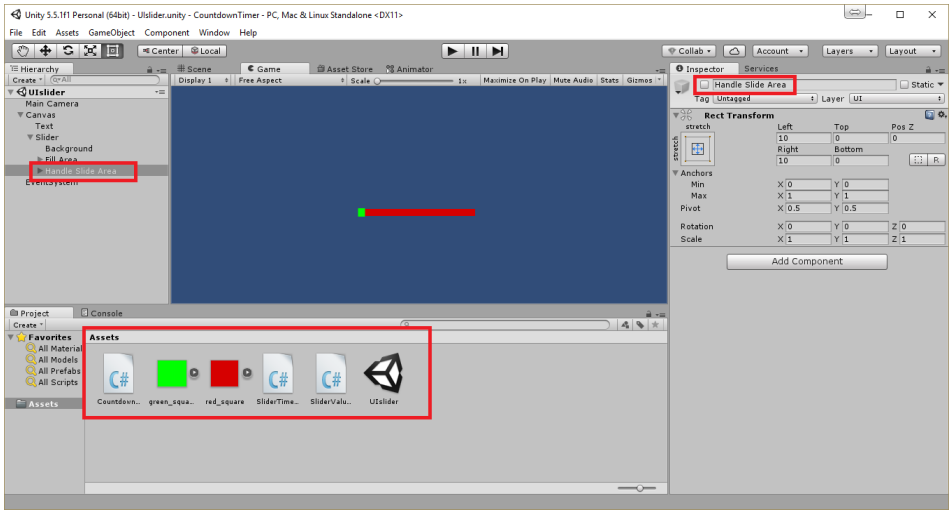
10. Save dan run (ctrl+b).
11. Hasil output :



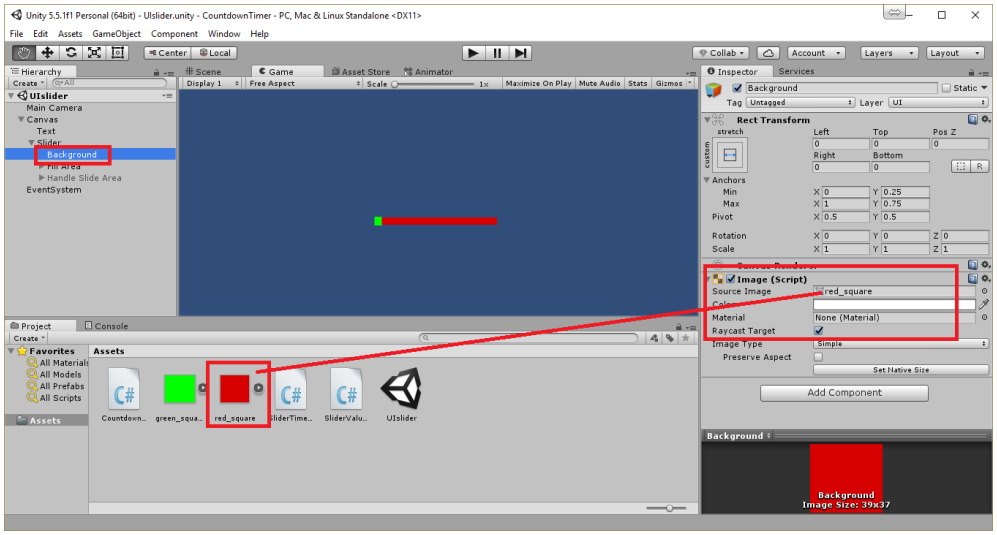
Ketika slider digerakkan, maka nilai pada teks akan diupdate dan ditambilkan angkanya, dimana min value = 0 dan max value = 20.

Membuat penghitung waktu (countdown timer) dengan menggunakan UI slider

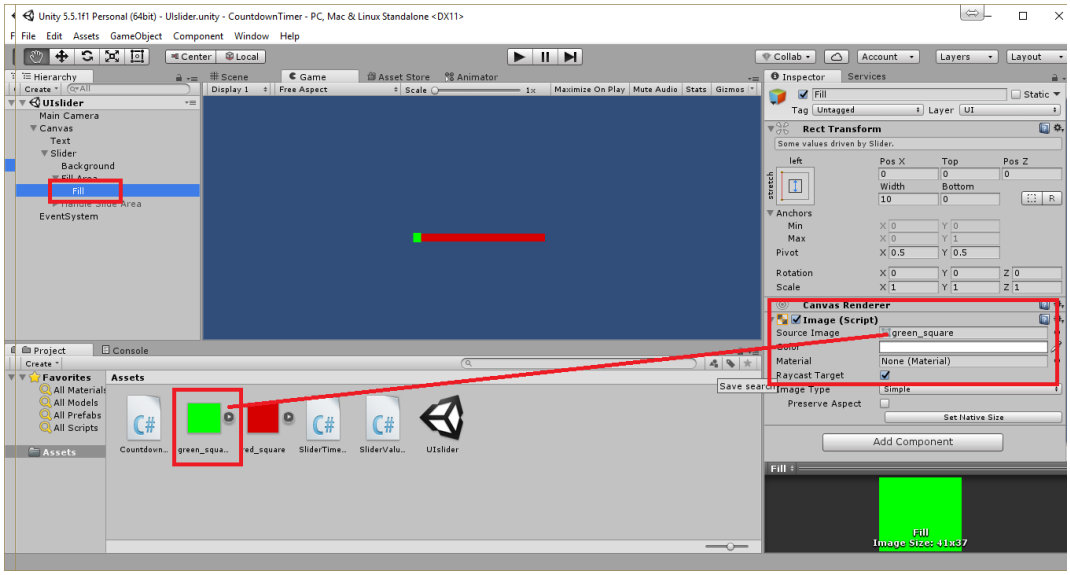
1. Untuk proses ini memerlukan **Asset** pada folder 1362_01_10.
2. Copy file **UI Slider** dari proses sebelumnya, kemudian rename file menjadi Countdown.
3. Masukkan asset yang diperlukan dari folder nomor 1, dengan cara drag script **Countdowntimer, gambar red_square, dan gambar green_square** kedalam tab **Asset** pada layar unity.
4. Pastikan GameObject **Slider** dipilih pada **Hierarchy.**
5. Non aktifkan Handle Slide Area dengan cara hapus centangnya.



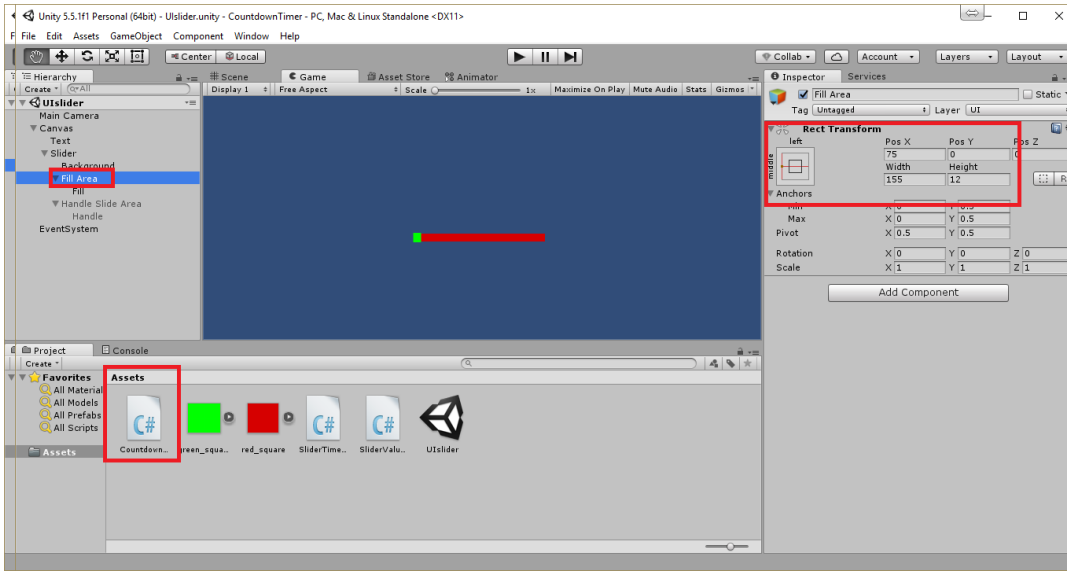
6. Pilih **Background** pada **Hierarchy.** Drag gambar **red_square** kedalam **Source Image** pada **Image (Script).**



7. Pilih **Fill** pada **Hierarchy**. Drag gambar **green_square** kedalam **Source Image** pada **Image (Script).**



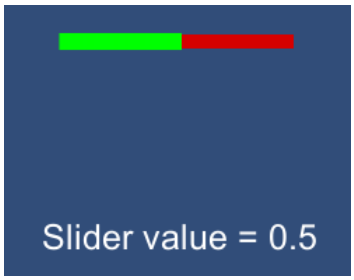
8. Pilih **Fill Area.** Pada komponen **Rect Transform,** gunakan posisi left-middle dan untuk **width** dan **height** adalah 155 dan 12.
9. Pastikan GameObject **Slider** dipilih pada **Hierarchy.** Masukkan script **CountTimerDisplay** dari folder **Asset** kedalam window **Asset** pada Unity.



10. Buatlah script C# dengan nama **SliderTimerDisplay** dengan source code dibawah ini. Kemudian drag kedalam GameObject **Slider.**

```
using System.Collections;
using UnityEngine.UI;
public class SliderTimerDisplay : MonoBehaviour
{
    private CountdownTimer countdownTimer;
    private Slider sliderUI;
    private int startSeconds = 30;
    void Start()
    {
        SetupSlider();
        SetupTimer();
    }
    void Update()
    {
        sliderUI.value =
        countdownTimer.GetProportionTimeRemaining();
        print(countdownTimer.GetProportionTimeRemaining());
    }
    private void SetupSlider()
    {
        sliderUI = GetComponent<Slider>();
        sliderUI.minValue = 0;
        sliderUI.maxValue = 1;
        sliderUI.wholeNumbers = false;
    }
    private void SetupTimer()
    {
        countdownTimer = GetComponent<CountdownTimer>();
        countdownTimer.ResetTimer(startSeconds);
    }
}
```

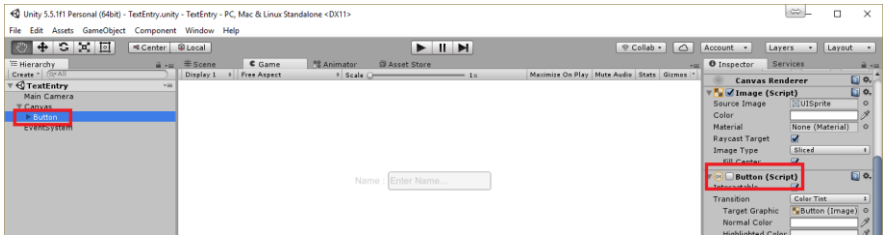
11. Save dan run (ctrl+b).
12. Hasil output :



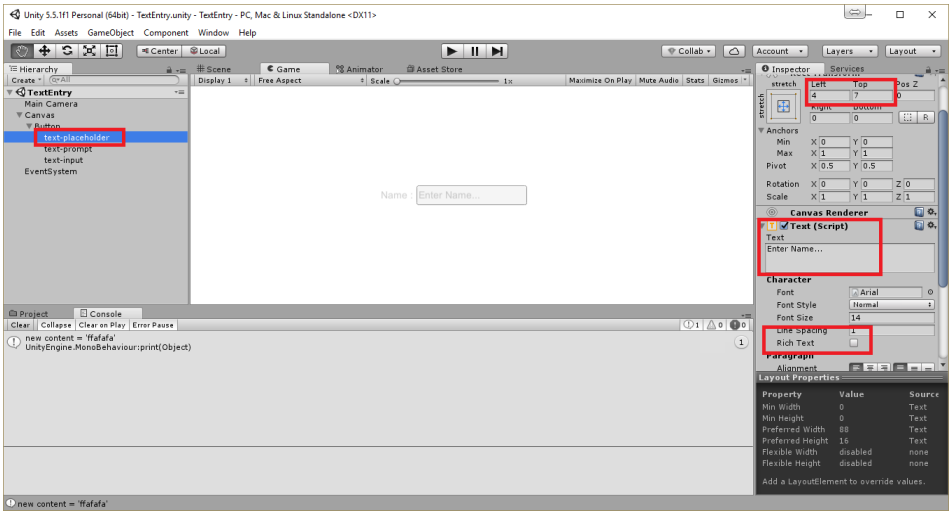
Slider akan otomatis berjalan dengan animasi warna hijau-merah, dan **Slider value** akan terupdate nilainya.

- **Membuat komponen input field untuk memasukkan teks (text entry)**

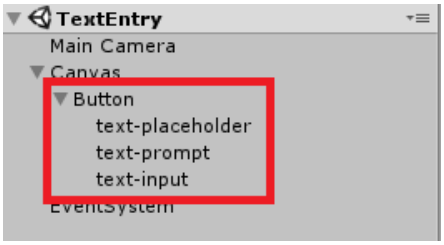
1. Buatlah projek 2D baru pada unity.
2. Pada **Inspector view,** ubahlah background dari **Main Camera** dengan warna putih.
3. Tambahkan **UI Button** ke layar. Hapus tanda centang pada **Button** di **Inspector view.**



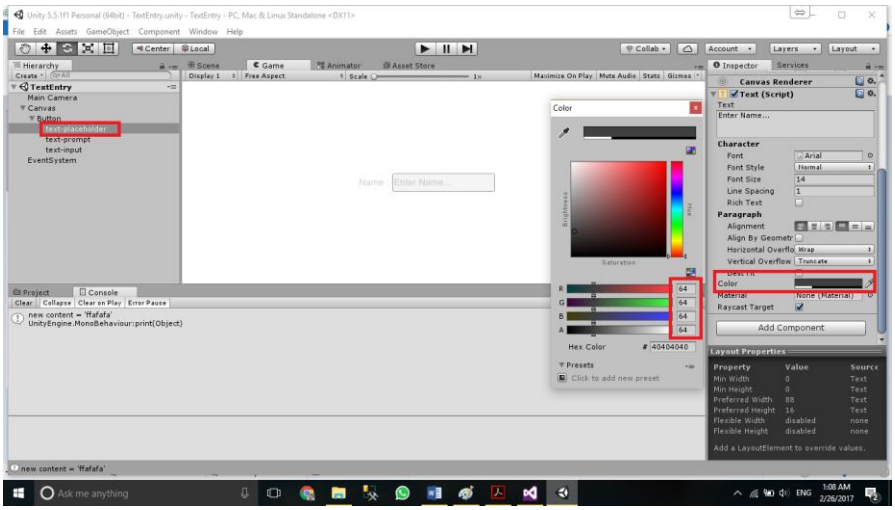
4. Ganti nama **Text** pada **Button** dengan **Text-placeholder**. Hapus tanda centang pada **Rich Text**. Edit isi teks menjadi **Enter name…** Gantikan **Alignment** untuk **Left** = 4 dan **Top** = 7.



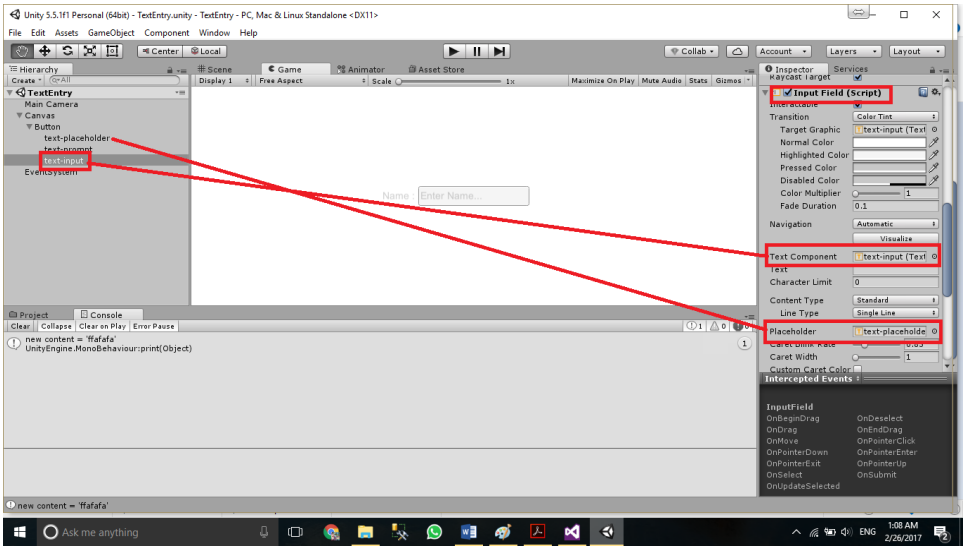
5. Copy objek **Text-placeholder**. Kemudian, berikan nama **Text-prompt.** Edit teks menjadi **Name : ,** kemudian atur posisi **Left = -50.**
6. Copy objek **Text-placeholder** lagi. Kemudian berikan nama **Text-input.** Hapus isi teks.



7. Pilih **Text-placeholder** pada **Hierarchy.** Ganti warna dari **Text(Script)** untuk (R,G,B,A) → (64,64,64,64).



8. Pilih **Text-input** pada **Hierarchy**. Tambahkan komponen **Input Field** dengan cara **Add Component → UI → Input Field**.
9. Pilih **Text-input** pada **Hierarchy**. Drag GameObject **Text-input** kedalam **Text Component** dari **Input Field(Script),** kemudian drag **Text-placeholder** kedalam **Placeholder.**



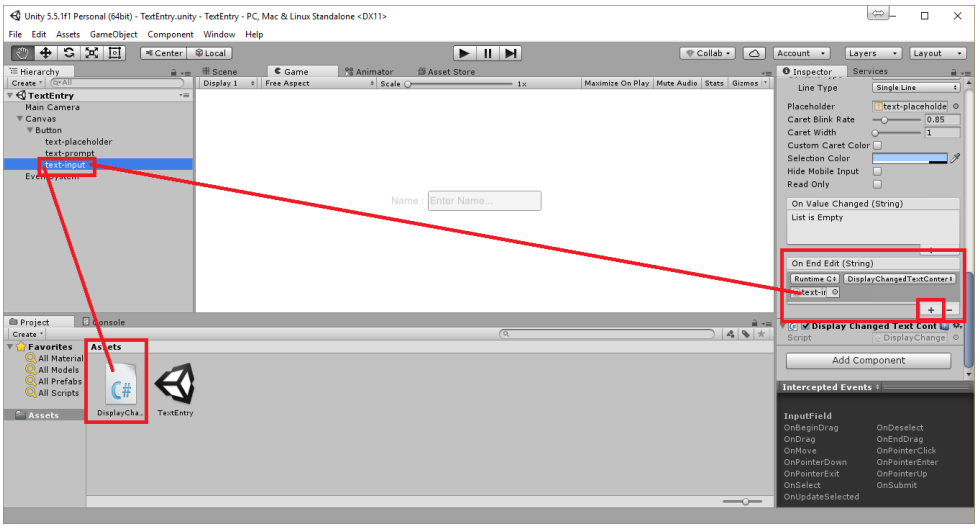
10. Save dan run (ctrl+b).
11. Hasil output :

✓ Tambahan : untuk menampilkan hasil respon dari setiap perubahan inputan user, pada **console**, ikutilah step selanjutnya (12→15).

12. Tambahkan script C# dengan nama DisplayChangedTextContent kedalam GameObject **Text-input,** dimana source code seperti berikut ini.

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;
public class DisplayChangedTextContent : MonoBehaviour
{
    private InputField inputField;
    void Start()
    {
        inputField = GetComponent<InputField>();
    }
    public void PrintNewValue()
    {
        string msg = "new content = '" + inputField.text + "'";
        print(msg);
    }
}
```

13. Pilih **Text-input** pada **Hierarchy**. Tambahkan event **End Edit (String)** untuk **Input Field (Script).** Klik tombol plus "+", kemudian drag **Text-input** kedalam **Object.** Pilih method **DisplayChangedTextContent,** kemudian pilih fungsi **PrintNewValue.**

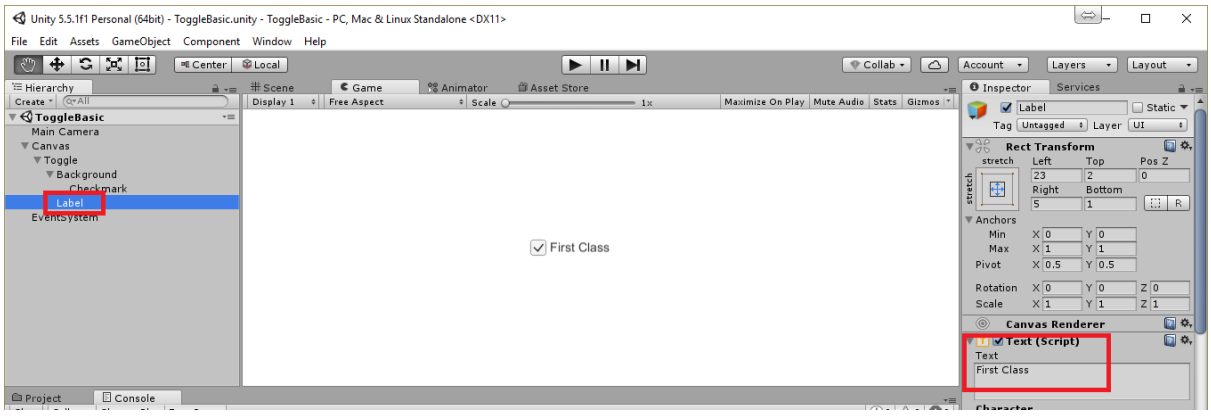


14. Save dan run (ctrl+b).
15. Hasil output :

Ketik nama sebagai inputan user, kemudian tekan tombol **Enter.** Buka tab **Console** untuk mengetahui hasil respon dari sistem jika terjadi suatu inputan yang baru.

Membuat toggle basic

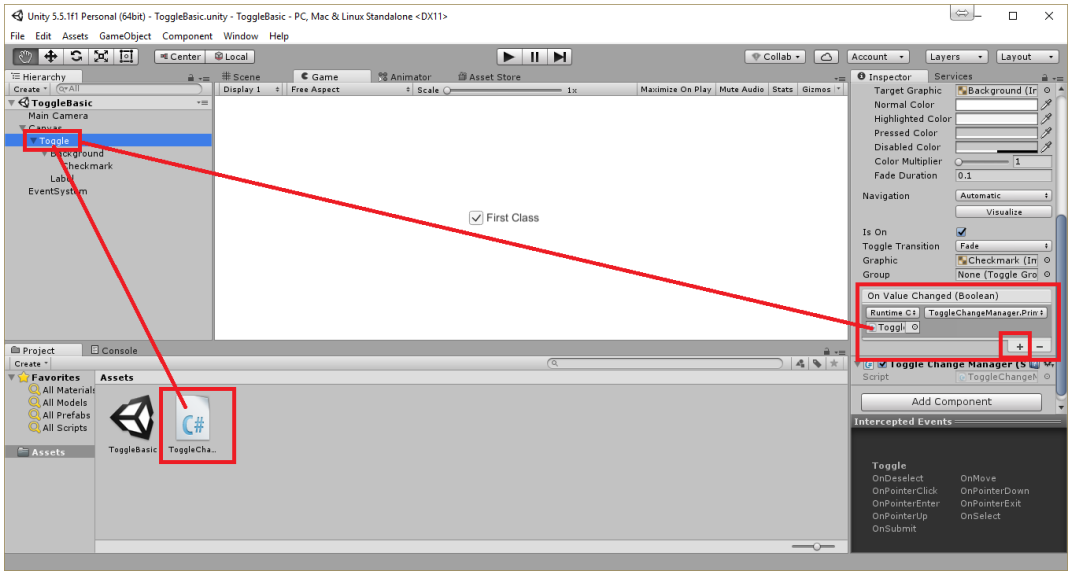
1. Untuk proses ini memerlukan **Asset** pada folder 1362_01_15.
2. Buatlah projek 2D yang baru pada unity.
3. Pada **Inspector panel,** ganti warna **Background** dari **Main Camera** menjadi putih.
4. Tambahkan **UI Toggle** kedalam layar **(GameObject→UI→Toggle).**
5. Expand GameObject **Toggle** pada **Hierarchy,** kemudian pilih **Label.** Edit teks menjadi **First Class.**



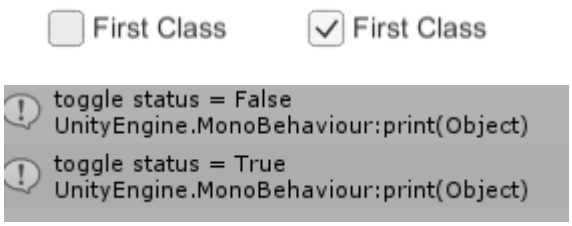
6. Buatlah script C# dengan nama **ToggleChangeManager** dengan source code seperti dibawah ini, kemudian masukkan kedalam GameObject **Toggle**.

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class ToggleChangeManager : MonoBehaviour
{
    private Toggle toggle;
    void Start()
    {
        toggle = GetComponent<Toggle>();
    }
    public void PrintNewToggleValue()
    {
        bool status = toggle.isOn;
        print("toggle status = " + status);
    }
}
```

7. Pilih GameObject **Toggle.** Tambahkan event **On Value Changed** pada **Toogle(Script).** Klik tombol plus "+", kemudian drag **Toggle** kedalam bagian **Object**. Pilih method **ToggleChangeManager**, kemudian pilih fungsi **PrintNewToggleValue.**



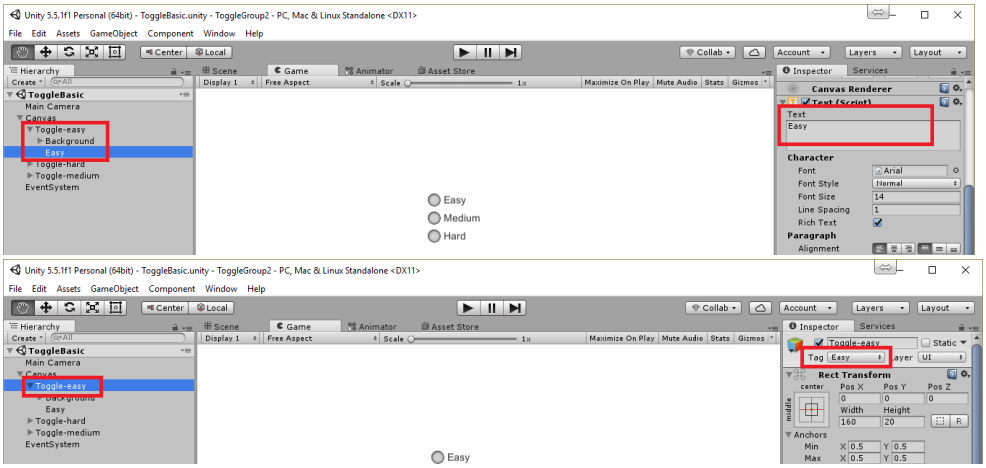
8. Save dan run (ctrl+b).
9. Hasil output :



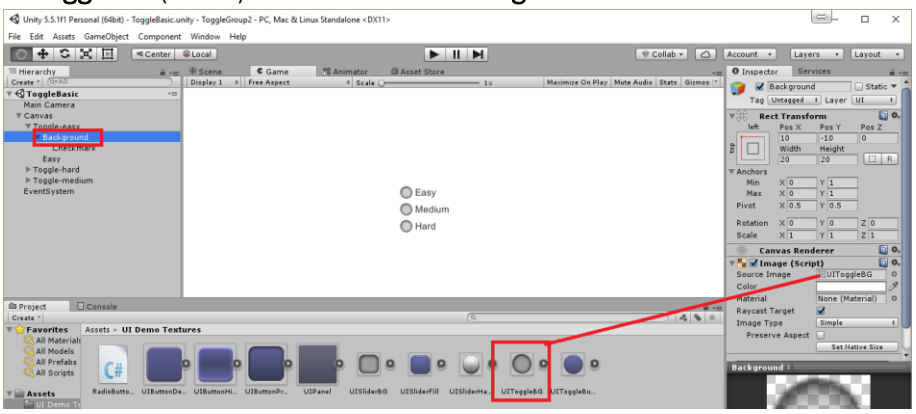
Jika **Toggle** dicentang, maka unity akan memberikan respon **True.**
Jika **Toggle** non-centang, maka unity akan memberikan respon **False.**

Membuat radio buttons dengan meggunakan toggle group

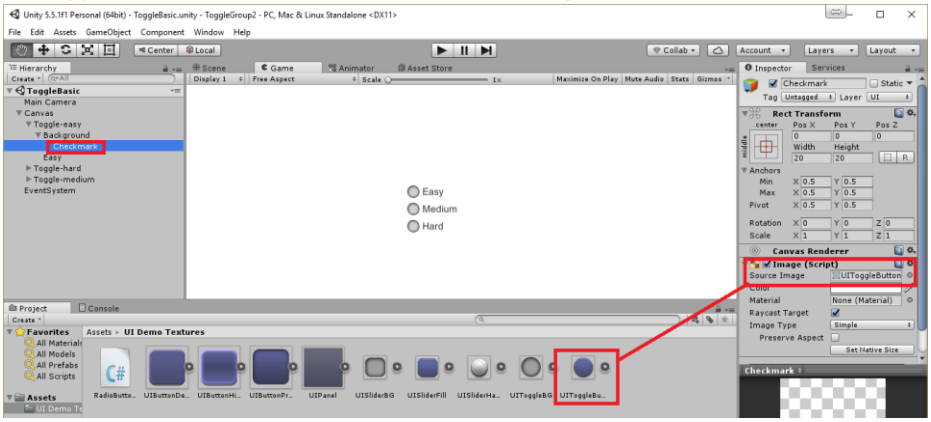
1. Untuk proses ini memerlukan **Asset** pada folder 1362_01_15.
2. Copy file dari projek **toggle basic** sebelumnya.
3. Import folder **UI Demo Textrures** dari Asset folder 1362_01_15.
4. Hapus script C# **ToggleChangeManager** dari GameObject **Toogle.**
5. Ganti nama GameObject **Toggle** menjadi **Toggle-easy.**
6. Expand **Toogle-easy**, kemudian ganti nama **Label** menjadi **Easy.**
7. Edit teks pada **Easy** menjadi **Easy,** kemudian ganti **tag** untuk GameObject ini dengan **tag** baru yang bernama **Easy.**



8. Expand **Toogle-easy,** kemudian pilih **Background.** Pada **Image(Script),** drag gambar **UIToggleBG** (Asset) kedalam **Source Image.**



9. Expand **Toogle-easy,** kemudian pilih **Checkmark.** Pada **Image(Script),** drag gambar **UIToggleButton** (Asset) kedalam **Source Image.**



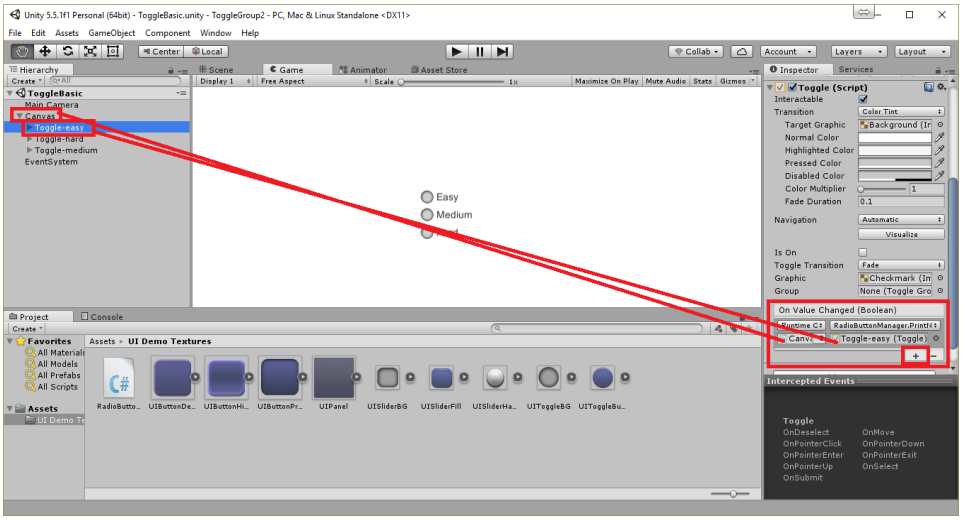
10. Copy GameObject **Toggle-easy,** kemudan edit hasil copy dengan nama **Toggle-medium.** Aturlah **Rect Transform** dimana **Pos Y = -25**. Ganti **tag** pada GameObject ini dengan **tag** baru dengan nama **Medium**.
11. Copy GameObject **Toggle-medium,** kemudan edit hasil copy dengan nama **Toggle-hard.** Aturlah **Rect Transform** dimana **Pos Y = -50**. Ganti **tag** pada GameObject ini dengan **tag** baru dengan nama **Hard**.
12. Buatlah script C# dengan nama **RadioButtonManager** seperti **source code** dibawah ini. Kemudian masukkan kedalam GameObject **Canvas.**

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;
public class RadioButtonManager : MonoBehaviour
{
    private string currentDifficulty = "Easy";
    private Toggle toggleEasy;
    private Toggle toggleMedium;
    private Toggle toggleHard;


    private void Start()
    {
        // Get all toggle
        toggleEasy = transform.Find("Toggle-easy").GetComponent<Toggle>();
        toggleMedium = transform.Find("Toggle-medium").GetComponent<Toggle>();
        toggleHard = transform.Find("Toggle-hard").GetComponent<Toggle>();
    }
    public void PrintNewGroupValue(Toggle sender)
    {
        // only take notice from Toggle just swtiched to On
        if (sender.isOn)
        {
            currentDifficulty = sender.tag;
            print("option changed to = " + currentDifficulty);

            //if sender is easy, so hard and medium are off
            if(sender.tag == "Easy")
            {
                toggleMedium.isOn = false;
                toggleHard.isOn = false;
            }

            else if (sender.tag == "Medium")
            {
                toggleEasy.isOn = false;
                toggleHard.isOn = false;
            }

            else if (sender.tag == "Hard")
            {
                toggleEasy.isOn = false;
                toggleMedium.isOn = false;
            }
        }
    }
}
```

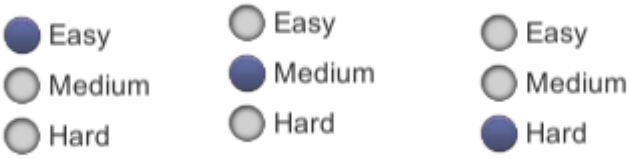
Logic for Radio Button

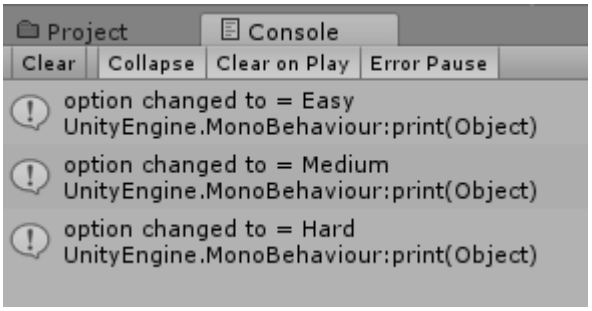
13. Pilih GameObject **Toggle-easy,** tambahkan event **On Value Changed** dari komponen **Toggle(Script).** Klik tombol plus “+”, kemudian drag GameObject **Canvas** kedalam

bagian **Object.** Pilih method **RadioButtonManager,** kemudian pilih fungsi **PrintNewGroupValue.** Pada bagian **None(Toggle),** drag GameObject **Toggle-easy** untuk menggantikannya.



14. Lakukan step 13 untuk **Toggle-medium** dan **Toggle-hard.**
15. Save dan run (ctrl+b).
16. Hasil output :





- ✓ Ketika tombol **Easy** ditekan, maka tombol **Medium** dan **Hard** akan mati.
- ✓ Ketika tombol **Medium** ditekan, maka tombol **Easy** dan **Hard** akan mati.
- ✓ Ketika tombol **Hard** ditekan, maka tombol **Easy** dan **Medium** akan mati.
- ✓ Unity akan memberikan respon untuk tombol yang ditekan seperti yang ditampilkan pada bagian **Console.**

F. TUGAS PRAKTIKUM

Buatlah User Interface (UI) game yang menggunakan semua komponen dari modul 2. Buat 3 layar (scene) yang saling berhubungan satu dengan yang lainnya. Tampilkan di desktop (.exe).

--- SELAMAT BELAJAR ---