

USER INPUT

A. JENIS-JENIS USER INPUT

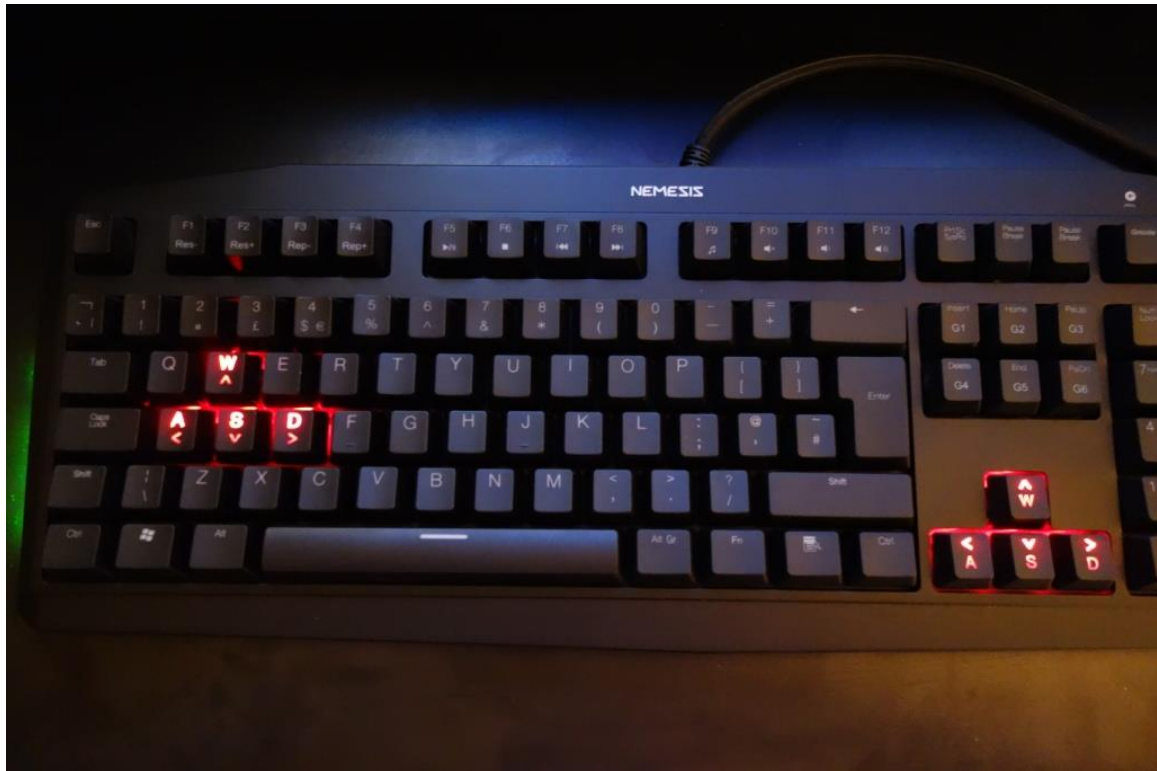
Ada beberapa jenis user input yang paling sering digunakan, antara lain:

- a.) Key arrow
- b.) Keyboard
- c.) Touch
- d.) Hover
- e.) Event System : Drag and Drop

Kita akan mencoba membahas satu persatu user input diatas terhadap gameobject

B. Key Arrow

Key Arrow yang dimaksud adalah tombol Kiri, Kanan, Atas, dan Bawah serta W,S,A,D. Posisinya pada keyboard adalah sebagai berikut:



Untuk bisa menggerakkan gameobject menggunakan key arrow diatas, adalah menggunakan syntaks `Input.GetAxis ("Horizontal")`

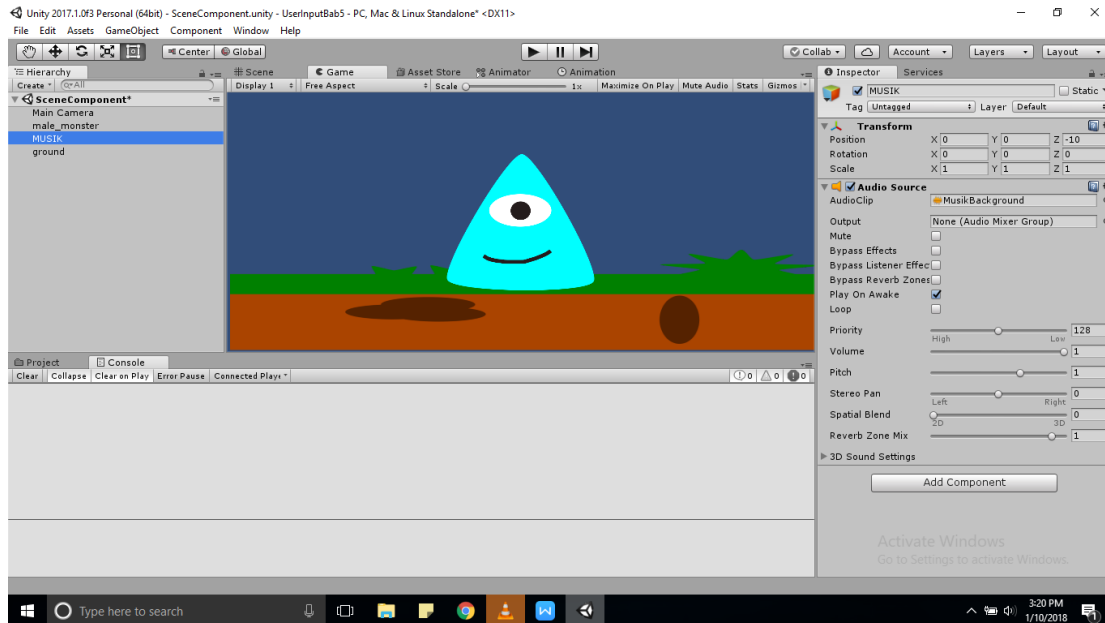
Untuk menggerakkan gameobject kekiri dan kekanan, dalam konteks ini adalah menekan tombol A,D, tombol kiri, dan tombol kanan.

`Input.GetAxis ("Vertical")`

Untuk menggerakkan gameobject keatas dan kebawah, dalam konteks ini adala dengan menekan tombol W,S, tombol atas, dan tombol bawah

Untuk uji menggerakkan karakter key arrow maka kita lakukan tahapan berikut:

Pertama kita siapkan sebuah scene menggunakan proyek unity yang sebelumnya



Lalu siapkan script bernama Gerak.cs sebagai berikut:

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

public class Gerak : MonoBehaviour {

    public float speed;

    // Use this for initialization
    void Start () {

    }

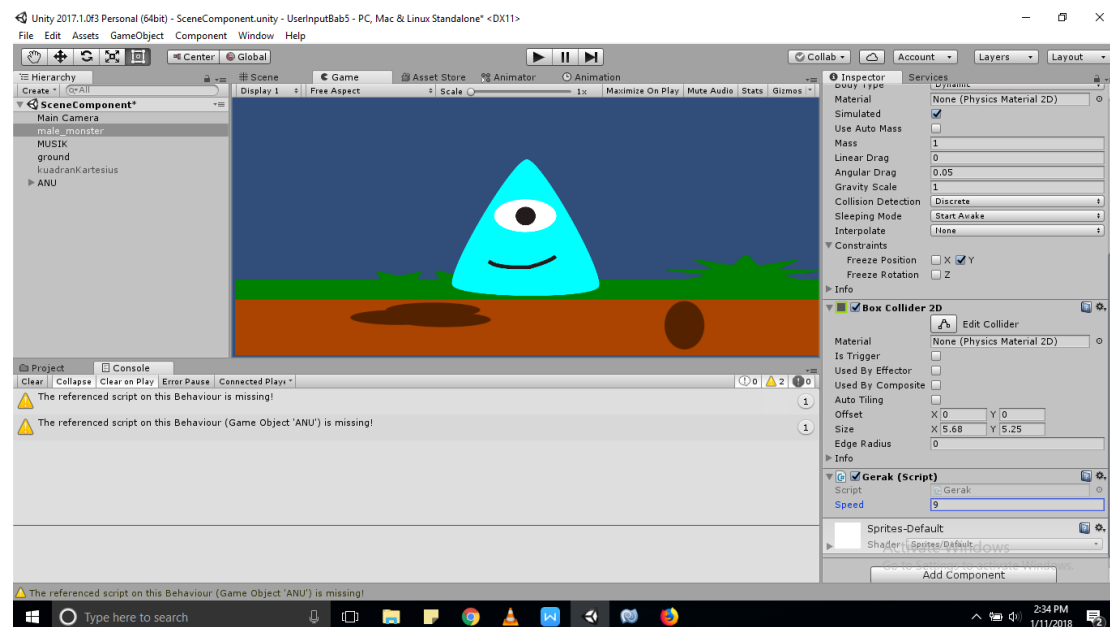
    // Update is called once per frame
    void Update () {
        if (Input.GetAxis ("Horizontal") != 0)
            transform.Translate (Vector2.right * speed * Input.GetAxis ("Horizontal") * Time.deltaTime);

        if (Input.GetAxis ("Vertical") != 0)
            transform.Translate (Vector2.up * speed * Input.GetAxis ("Vertical") * Time.deltaTime);
    }
}
```

Koding diatas bermakna bahwa setiap detik unity mengecek sebanyak fps game (misal jika game kita 30 fps maka koding diatas akan 30 kali perdetik) inputan user pada key arrow. Jika user terdeteksi menekan key arrow horizontal yaitu A, D, tombol kiri, dan tombol kanan maka unity akan mengeksekusi pengkondisian pertama yaitu menggerakkan karakter ke arah kanan sebanyak speed perdetik dikalikan faktor `Input.GetAxis("Horizontal")`. Jika tombol kekanan (D dan tombol kanan) ditekan maka `Input.GetAxis("Horizontal")` akan mengembalikan nilai 1 sehingga gameobject akan bergerak ke atas, sedangkan jika tombol ke kiri ditekan (A dan tombol kiri) maka `Input.GetAxis("Horizontal")` akan mengembalikan nilai -1 sehingga pergerakan `Vector2.right` yang seharusnya kekanan akan berperilaku sebaliknya yaitu ke kiri karena translasinya dinegasikan oleh angka minus 1.

Begitu pula jika user terdeteksi menekan key arrow vertikal yaitu W,S, tombol atas, dan tombol bawah maka unity akan mengeksekusi pengkondisian kedua yaitu menggerakkan karakter kearah atas sebanyak speed perdetik dikalikan faktor `Input.GetAxis("Vertical")`. Jika tombol keatas (W dan tombol atas) ditekan maka `Input.GetAxis("Vertical")` akan mengembalikan nilai 1 sehingga gameobject akan bergerak keatas, sedangkan jika tombol kekiri ditekan (S dan tombol bawah) maka `Input.GetAxis("Vertical")` akan mengembalikan nilai -1 sehingga pergerakan `Vector2.up` yang seharusnya keatas akan berperilaku sebaliknya yaitu kebawah karena translasinya dinegasikan oleh angka minus 1.

Silahkan dimainkan, jangan lupa untuk mengatur speed terlebih dahulu karena jika speed bernilai 0 maka gameobject tidak akan bergerak kemanapun. Dicoba juga semua tombol key Arrow yaitu WSAD dan tombol atas,bawah,kiri, dan kanan.



C. Keyboard

Keyboard yang dimaksud adalah semua tombol pada keyboard, termasuk key arrow yang kita bahas sebelumnya pun bisa jika kita koding dengan pendekatan input keyboard. Syntaks koding input keyboard adalah sebagai berikut:

`Input.GetKey`

Koding ini akan dieksekusi selama user menekan tombol sebanyak jumlah fps, cocok untuk digunakan untuk game shooter.



Game kapal raiden yang terus menembak peluru secara simultan jika tombol ditekan.

Input.GetKeyDown

Koding ini akan dieksekusi hanya sekali saja ketika tombol ditekan, cocok untuk hampir semua aksi didalam game karena menimbulkan kesan responsif seperti meloncat, mengeluarkan jurus, dsb



Game mario yang meloncat sekali saja ketika tombol ditekan.

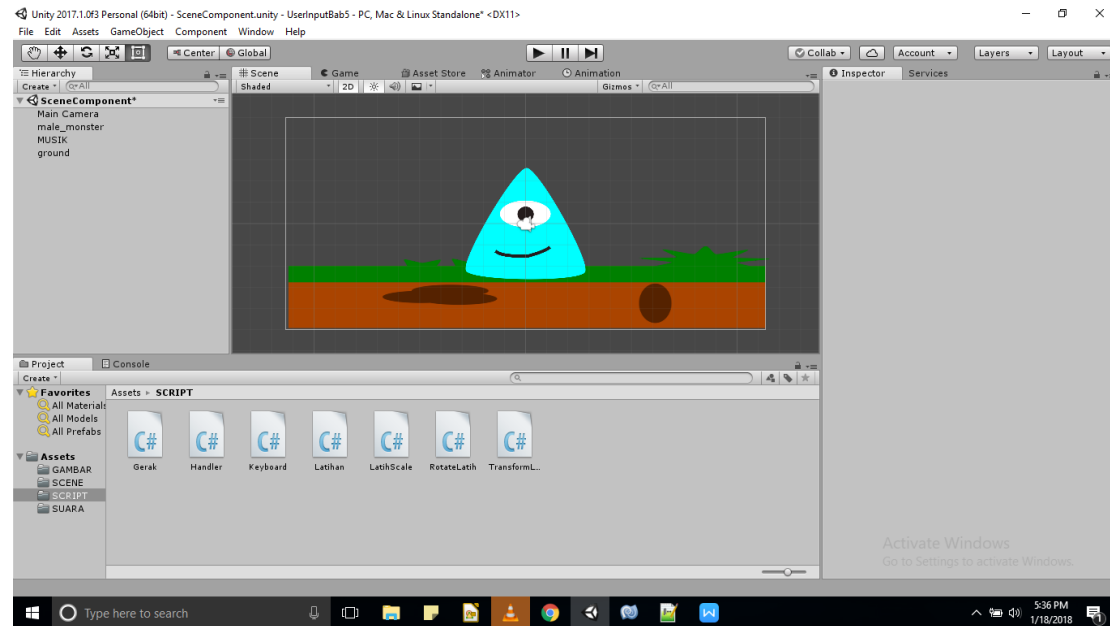
Input.GetKeyUp

Koding ini akan dieksekusi hanya sekali saja ketika sebuah tombol yang ditekan dilepaskan, cocok untuk hal-hal yang berkaitan dengan timing dan pengumpulan tenaga seperti mengumpulkan kekuatan dan baru ditembakkan ketika melepas tombol.

Baik sekarang kita, akan mengimplementasikan pada game

Input.GetKey

Untuk mempelajari Input.GetKey kita akan mengimplementasikan dengan cara saat menekan tombol maka gameobject kita akan berputar. Pertama siapkan scene sebagai berikut:



Buat script putar, diman kita akan menggabungkan koding Input.GetKey dengan koding rotasi yang kita pelajari kemarin. Dalam contoh akan digunakan spacebar sebagai key untuk memutar, silahkan kalian pilih tombol sendiri. Berikut adalah kodingnya.

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

public class Putar : MonoBehaviour {

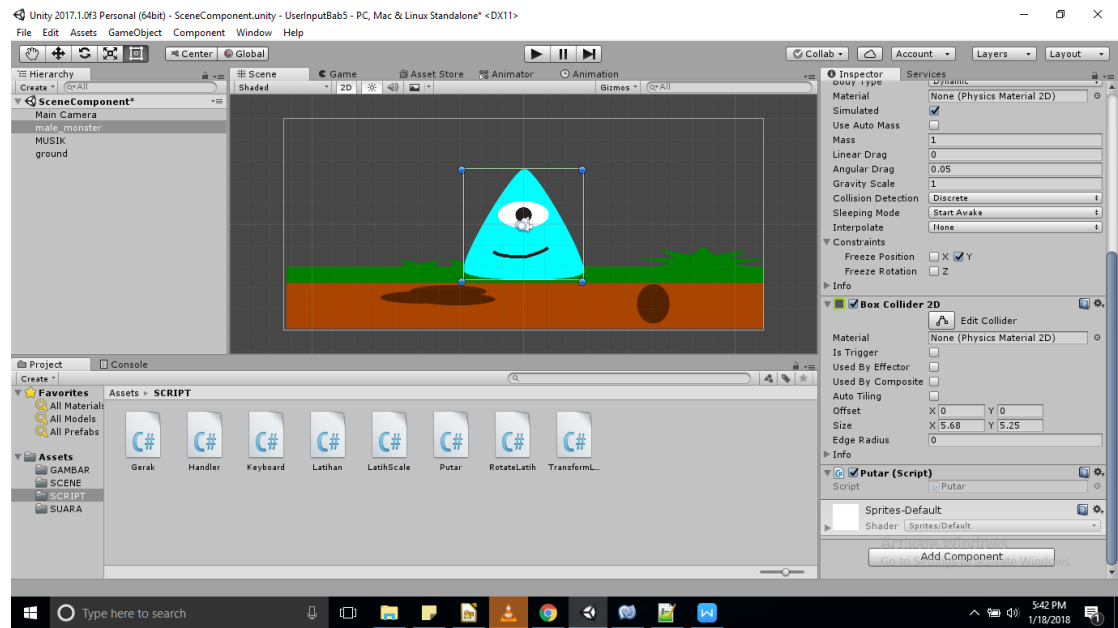
    // Use this for initialization
    void Start () {

    }

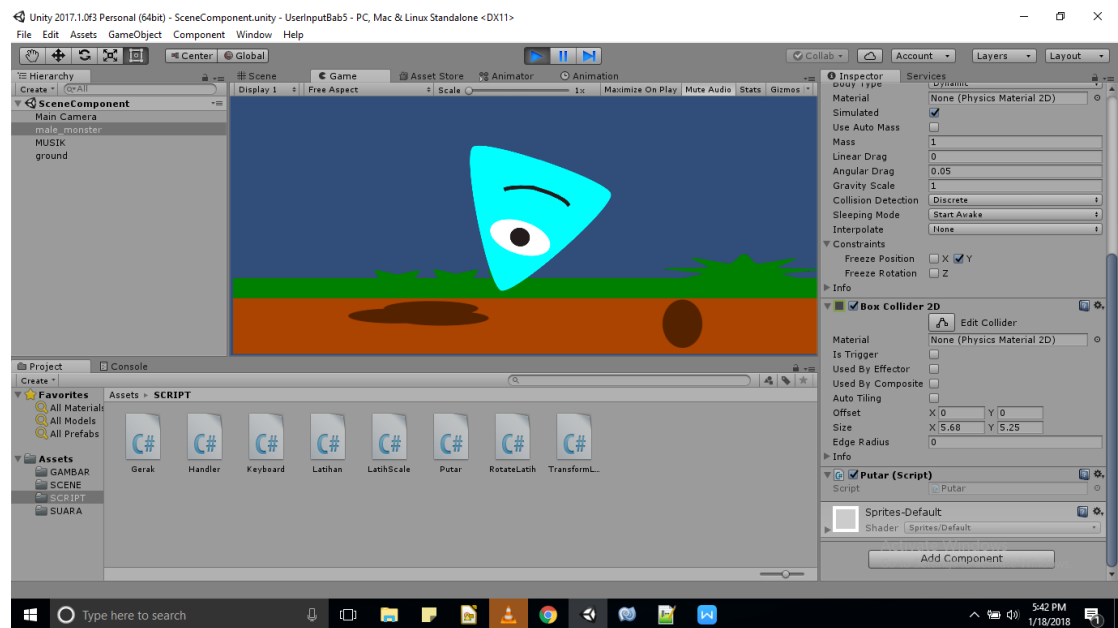
    // Update is called once per frame
    void Update () {
        pencet ();
    }
}
```

```
void pencet(){
    if (Input.GetKey (KeyCode.Space)) {
        transform.Rotate (0,0,3f);
    }
}
```

Hilangkan script2 sebelumnya(jika ada) pada gameobject monster lalu tambahkan script diatas pada gameobject monster.



Jika dijalankan lalu tekan tombol spasi atau tombol lain yang kalian pilih, hasilnya adalah:

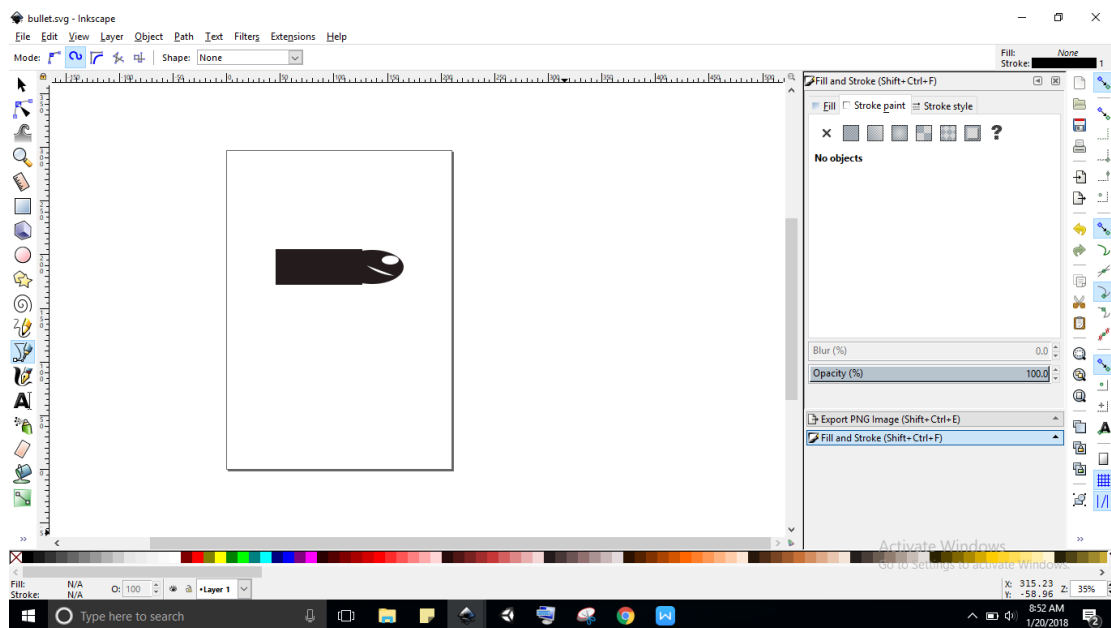


Silahkan dilepas maka gameobject monster pun akan berhenti berputar.

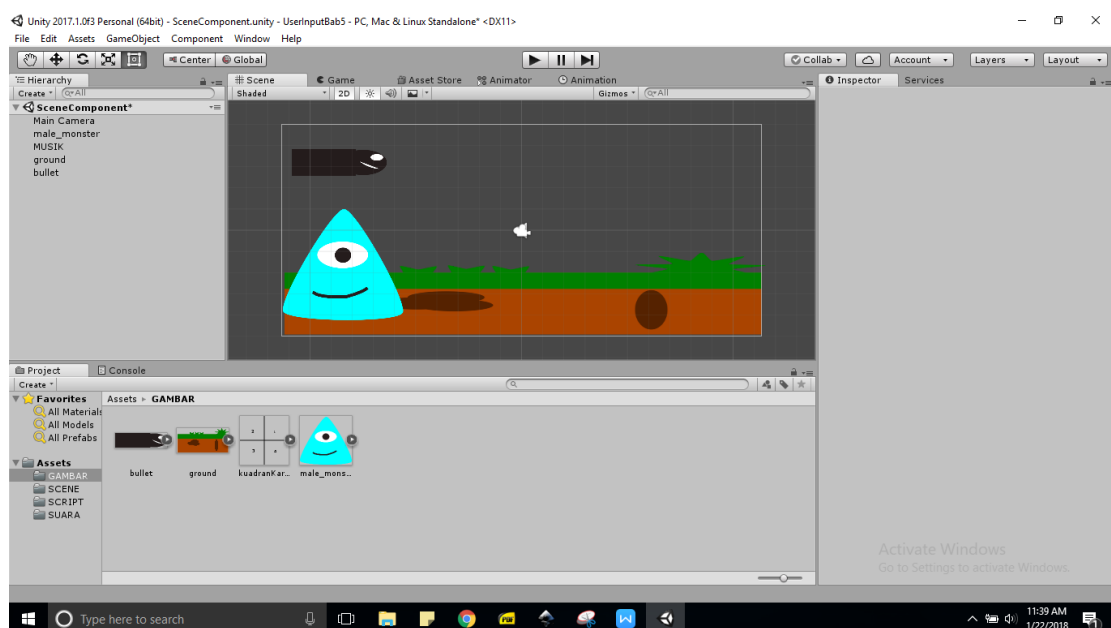
Baik sekarang kita, akan mengimplementasikan pada game

Input.GetKeyDown

Contoh kasus yang akan kita gunakan untuk mempelajari Input.GetKeyDown adalah menembak, pertama kita buat peluru dulu di inkscape seperti berikut:



Simpan peluru didalam scene dengan posisi sebagai berikut:



Buat script baru bernama move.cs sebagai berikut:

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

public class Move : MonoBehaviour {

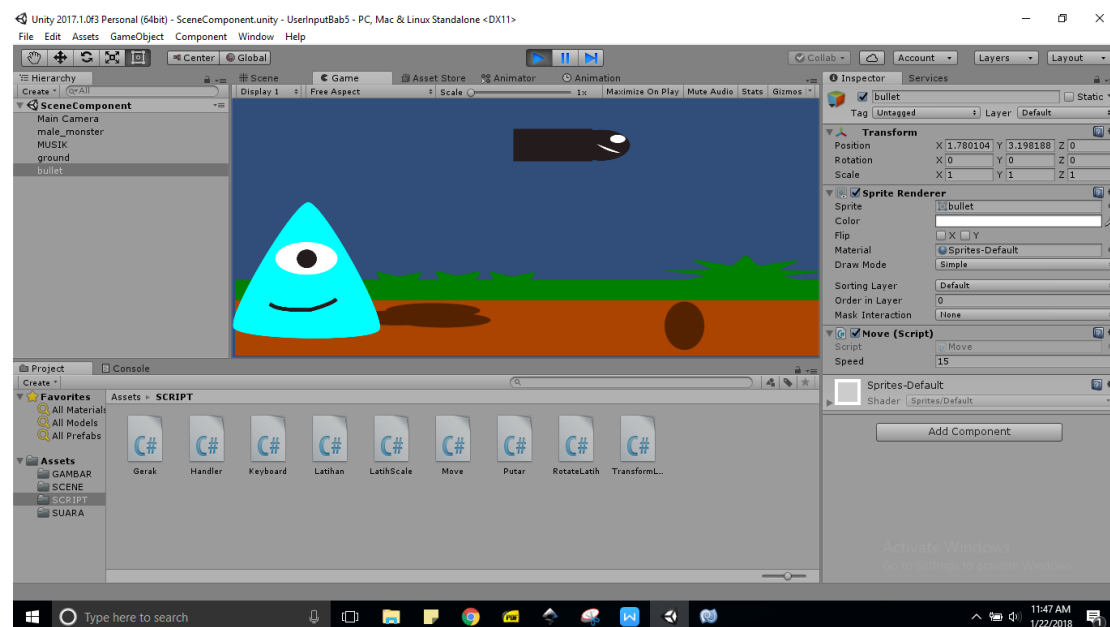
    public float speed;

    // Use this for initialization
    void Start () {

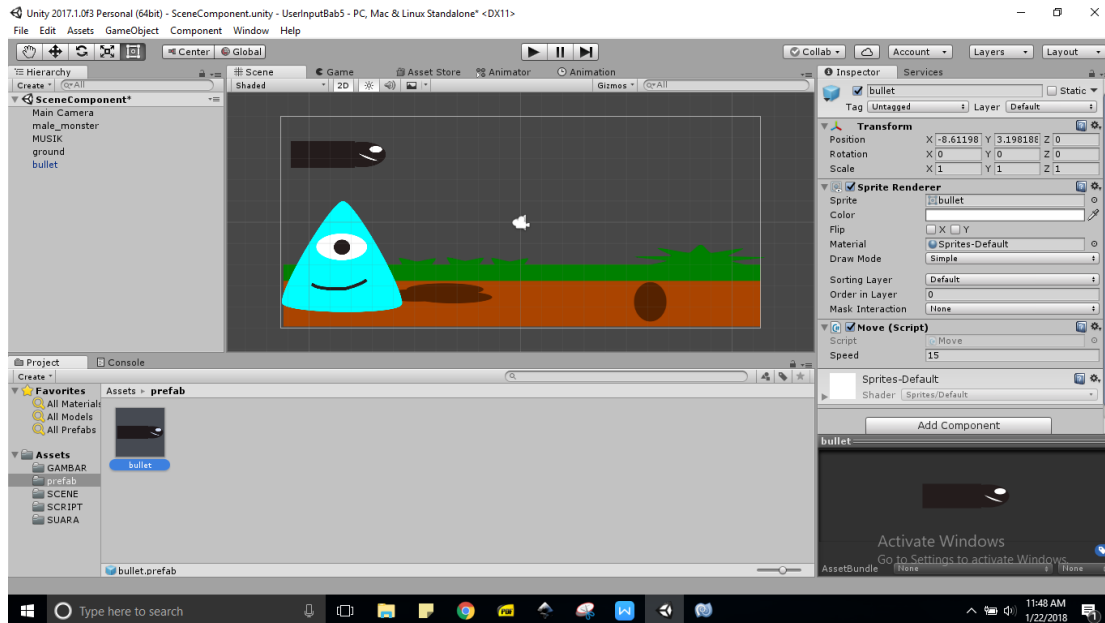
    }

    // Update is called once per frame
    void Update () {
        transform.Translate (Vector2.right * speed * Time.deltaTime);
    }
}
```

Jadikan script diatas sebagai component untuk peluru, atur kecepatannya lalu mainkan



Jadikan peluru tersebut sebagai prefab



Sekarang kita buat script untuk menembaknya, sebagai berikut:

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

public class Tembak : MonoBehaviour {

    // Use this for initialization
    void Start () {

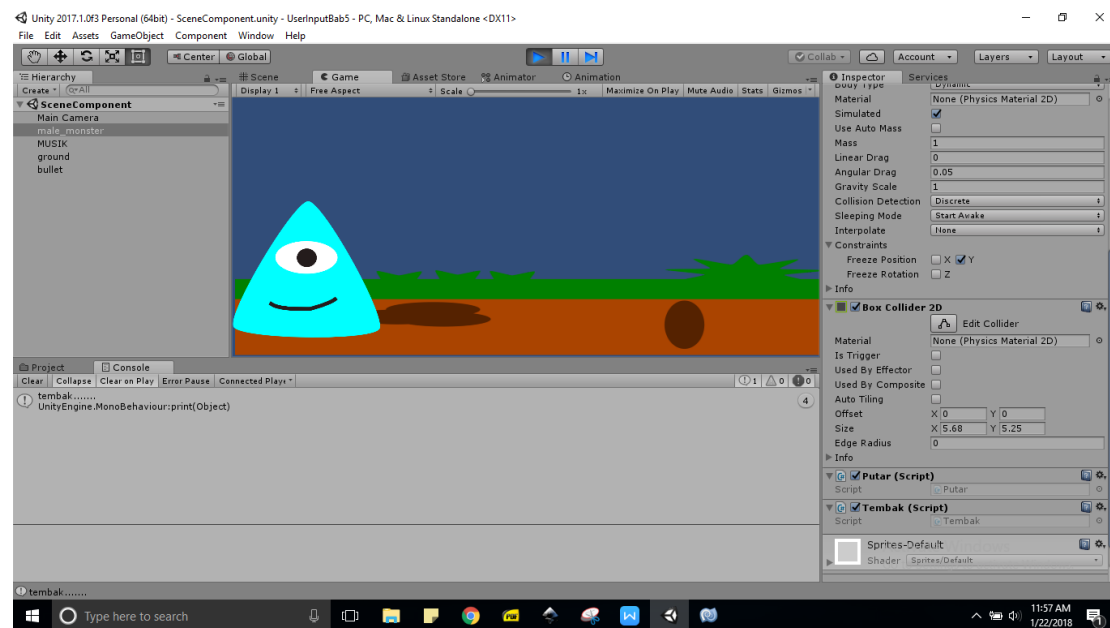
    }

    // Update is called once per frame
    void Update () {
        tembak ();
    }

    void tembak(){
        if(Input.GetKeyDown (KeyCode.Z)){
            print("tembak.....");
        }
    }
}
```

Tambahkan script Tembak.cs diatas sebagai component untuk monster, tekan tombol Z atau kode tombol yang lain untuk melihat tulisan yang muncul di console. Perhatikan bahwa jumlah tulisan

tembak yang muncul hanya akan muncul sekali dan tidak dieksekusi kembali walau tombol tsb ditekan selama tombol tersebut belum diangkat



Nah sekarang kita ingin agar jika kita tekan tombol maka peluru yang akan keluar, maka kita rombak kode di script tembak sebagai berikut:

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

public class Tembak : MonoBehaviour {

    public GameObject bullet; //A
    // Use this for initialization
    void Start () {

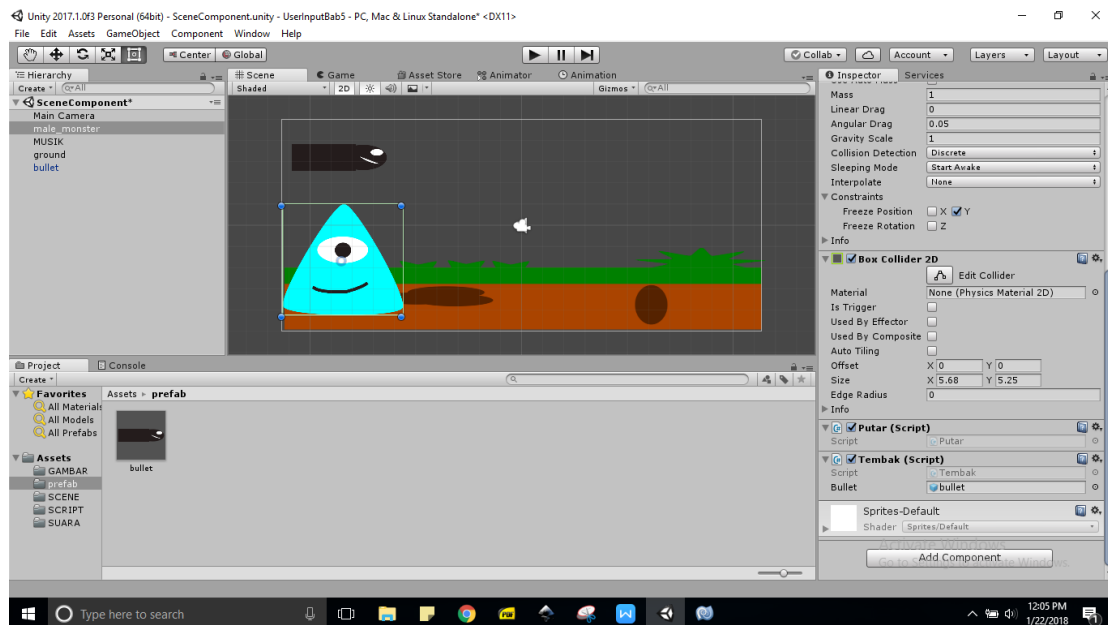
    }

    // Update is called once per frame
    void Update () {
        tembak ();
    }

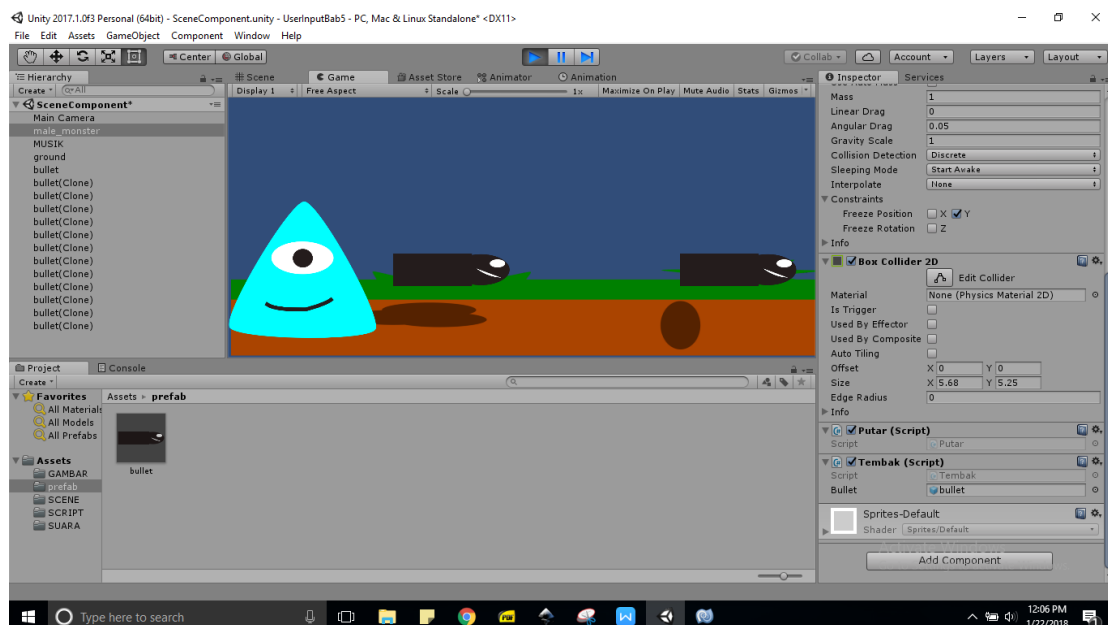
    void tembak(){
        if(Input.GetKeyDown (KeyCode.Z)){
            Instantiate (bullet, transform.position, Quaternion.identity); //B
        }
    }
}
```

Pada baris A, kita menambah gameobject public sehingga muncul di inspector dimana nanti kita akan isi dengan prefab peluru. Sedangkan pada baris B kita menambahkan kode untuk menciptakan

gameobject pada saat runtime (bukan dipersiapkan pada scene sebelum bermain), dimana saat kita tekan tombol maka peluru akan diciptakan sebagai gameobject pada scene. Dimana bullet adalah variabel gameobject yang kita jadikan publik diatas, transform position adalah dimana gameobject peluru akan diciptakan dalam hal ini gameobject peluru tersebut akan diciptakan dimana posisi gameobject yang memiliki komponen script ini yaitu monster, dan Quaternion.identity adalah kita memberitahu agar gameobject tidak dirotasi ketika diciptakan.



Sekarang tekan z untuk menembak pelurunya



Baik sekarang kita, akan mengimplementasikan pada game

`Input.GetKeyUp`

Skenarionya adalah kita akan mendorong peluru, semakin lama kita menekan tombol maka semakin kuat dorongan saat kita melepas tombol yang kita tekan. Untuk bisa menjalankan skenario pembelajaran Input.GetKeyUp ini maka kita perlu melakukan beberapa hal, pertama kita pastikan peluru memiliki box collider 2d, rigidbody 2d dynamic dan komponen script sebagai berikut.

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

public class ShootPower : MonoBehaviour {

    // Use this for initialization
    void Start () {

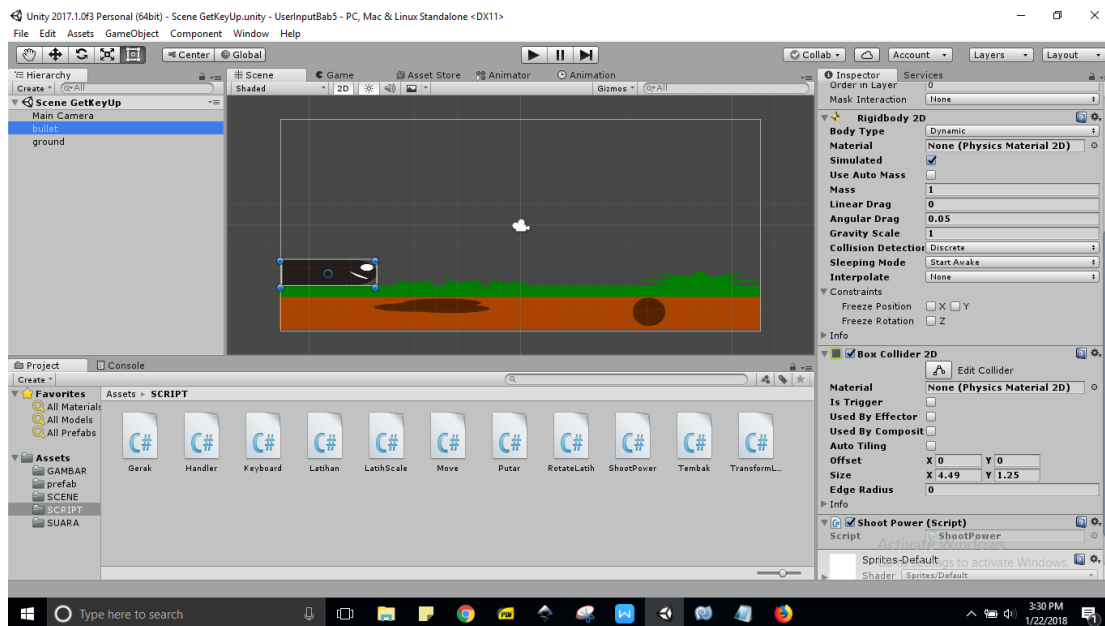
    }

    // Update is called once per frame
    void Update () {
        if(Input.GetKeyUp(KeyCode.Space)){ //A
            GetComponent<Rigidbody2D> ().AddForce (new Vector2(500f,0)); //B
        }
    }
}
```

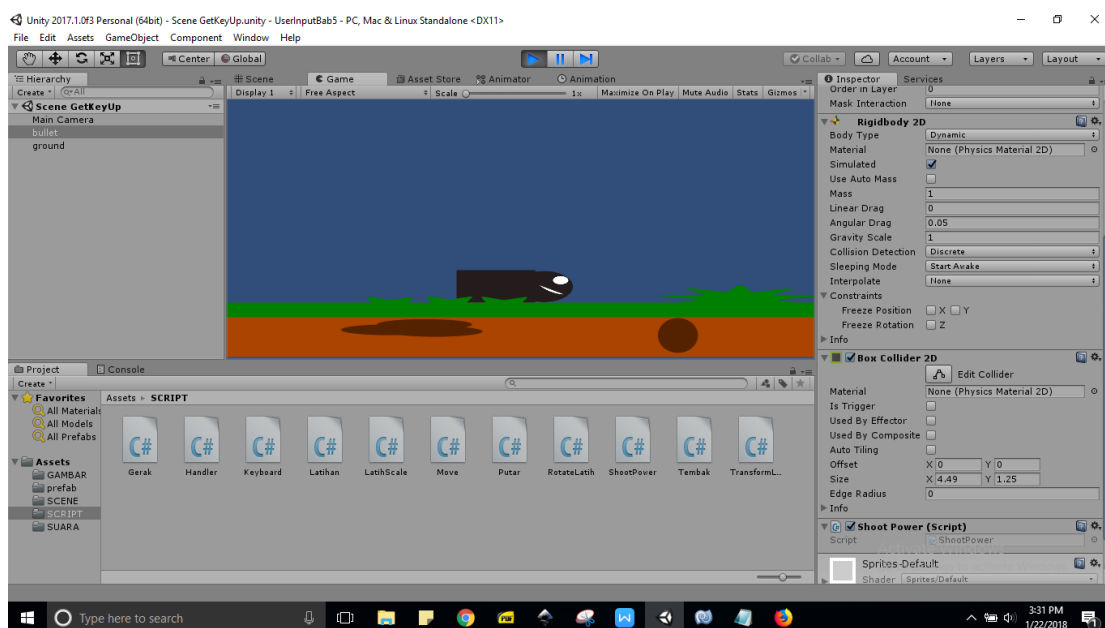
Pada script Shootpower diatas bagian A menjelaskan bahwa ketika kita melepas tombol yang sebelumnya ditekan maka akan melakukan aksi pada bagian B. Bagian B adalah kita memanggil komponen rigidbody2d untuk didorong. Syntax untuk mendorong gameobject yang memiliki rigidbody adalah AddForce sebagai berikut:

AddForce(new Vector(dorongan Vertikal, dorongan horizontal))

Pada bagian B diatas kita akan mencoba mendorong sebanyak 500 satuan horizontal kekanan (kalau ingin ke kiri nilainya minus) dan 0 satuan ke arah vertical. Posisikan scene anda yang baru sebagai berikut:



Mainkan, tekan spasi lalu lepas maka peluru anda akan terdorong



`Input.GetKeyUp` belum menarik sebelum kita mengimplementasikan penambahan power saat masih menekan dan baru melepaskan kekuatan saat kita melepaskan tombol. Jadi saat kita menekan

`Input.GetKeyDown` kita melakukan increment terhadap power xAxis yang akan diimplementasikan disini:

```
GetComponent<Rigidbody2D> ().AddForce (new Vector2(xAxis,yAxis));
```

Baru ketika kita lepas tombolnya maka AddForce diatas dieksekusi, maka script ShootPower dirubah sebagai berikut:

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

public class ShootPower : MonoBehaviour {

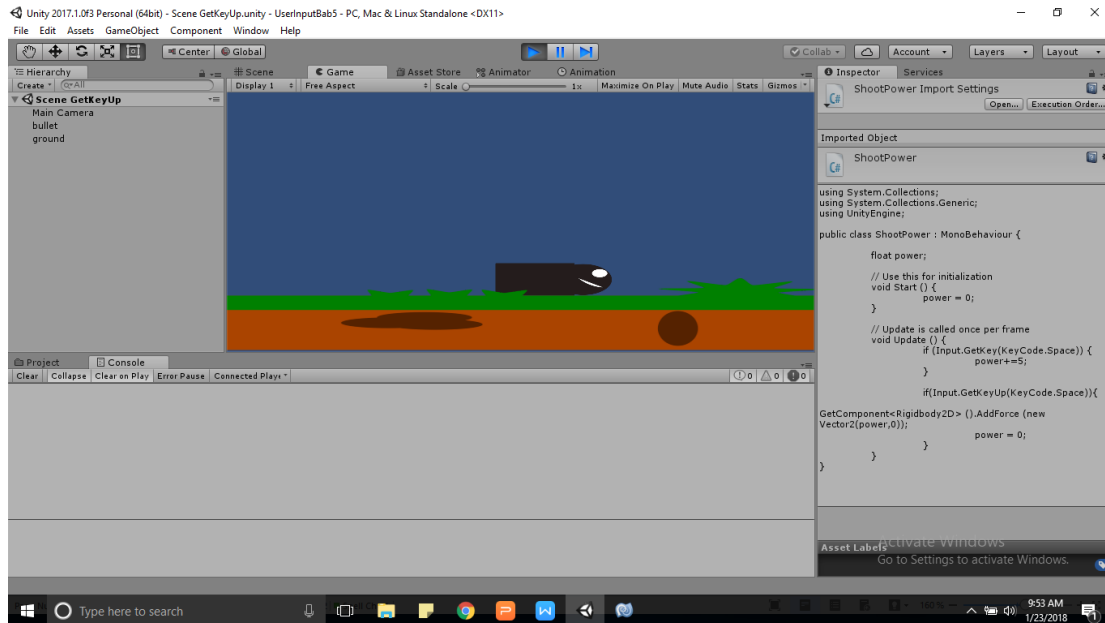
    float power; //A

    // Use this for initialization
    void Start () {
        power = 0;
    }

    // Update is called once per frame
    void Update () {
        if (Input.GetKey(KeyCode.Space)) { //B
            power+=5; //C
        }

        if(Input.GetKeyUp(KeyCode.Space)){
            GetComponent<Rigidbody2D> ().AddForce (new Vector2(power,0)); //D
            power = 0; //E
        }
    }
}
```

Perhatikan bahwa pada bagian A kita tambahkan variable power bertipe data float, dimana float pada bagian C akan diincrement 5 satuan per FPS selama tombol ditekan yang dikenali oleh unity lewat syntax GetKey di bagian B. Perhatikan bahwa kode yang ditekan dan dilepas harus sama. Baru pada saat tombol dilepas maka rigidbody2d akan kita dorong secara horizontal pada bagian D lalu kita 0 kan pada bagian E agar nanti ketika kita lepas maka power mulai dari 0 kembali. Silahkan dimainkan



D. Touch

Berikutnya interaksi yang sering dilakukan pemain adalah touch atau mengklik pada gameobject. Touch sebenarnya sangat sederhana cukup kita tambahkan script yang memiliki void sebagai berikut:

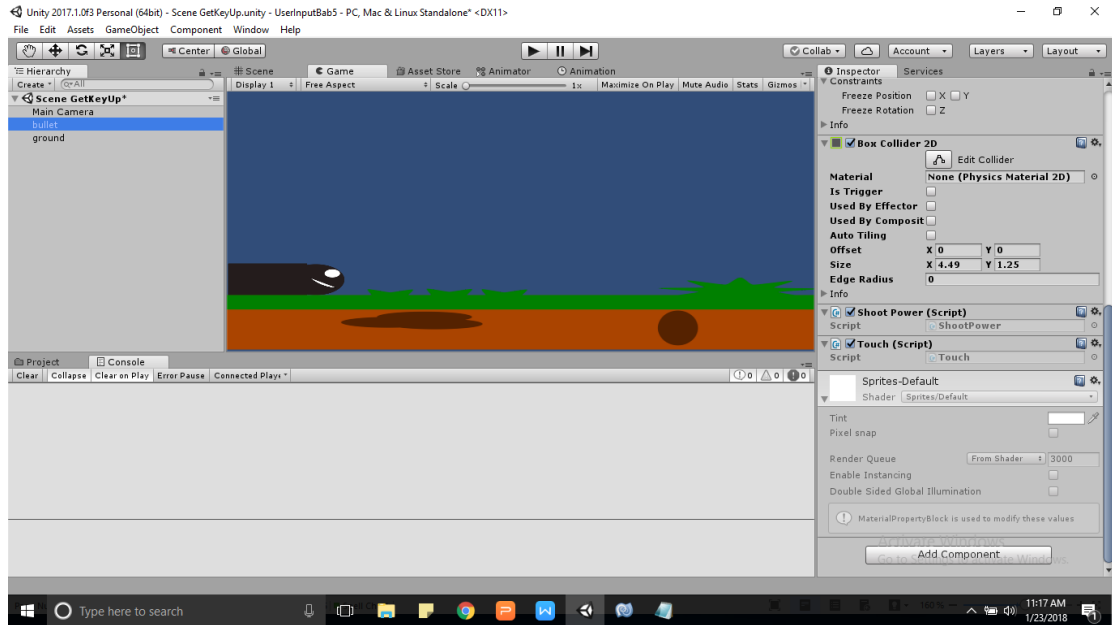
```
void OnMouseDown()  
{}
```

Misalkan kita buat script baru bernama Touch.cs sebagai berikut

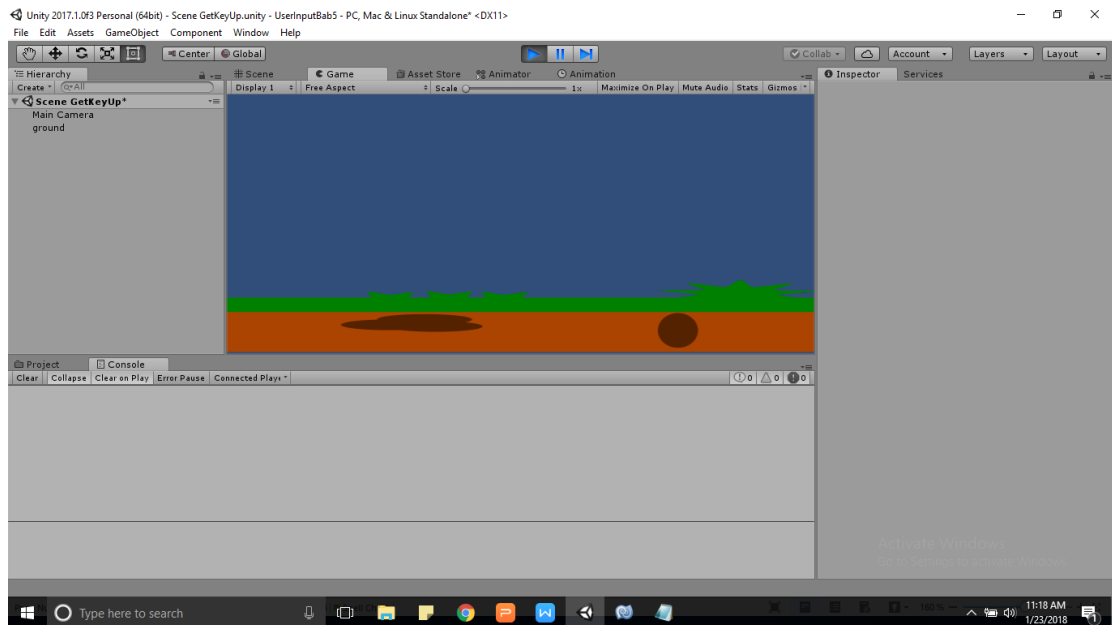
```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Touch : MonoBehaviour {  
  
    // Use this for initialization  
    void Start () {  
  
    }  
  
    // Update is called once per frame  
    void Update () {  
  
    }  
  
    void OnMouseDown()  
    {
```

```
        Destroy (gameObject);  
    }  
}
```

Tambahkan sebagai komponen pada peluru



Maka saat kita tekan peluru, peluru akan destroy/gameobjectnya dihapus



E. Hover

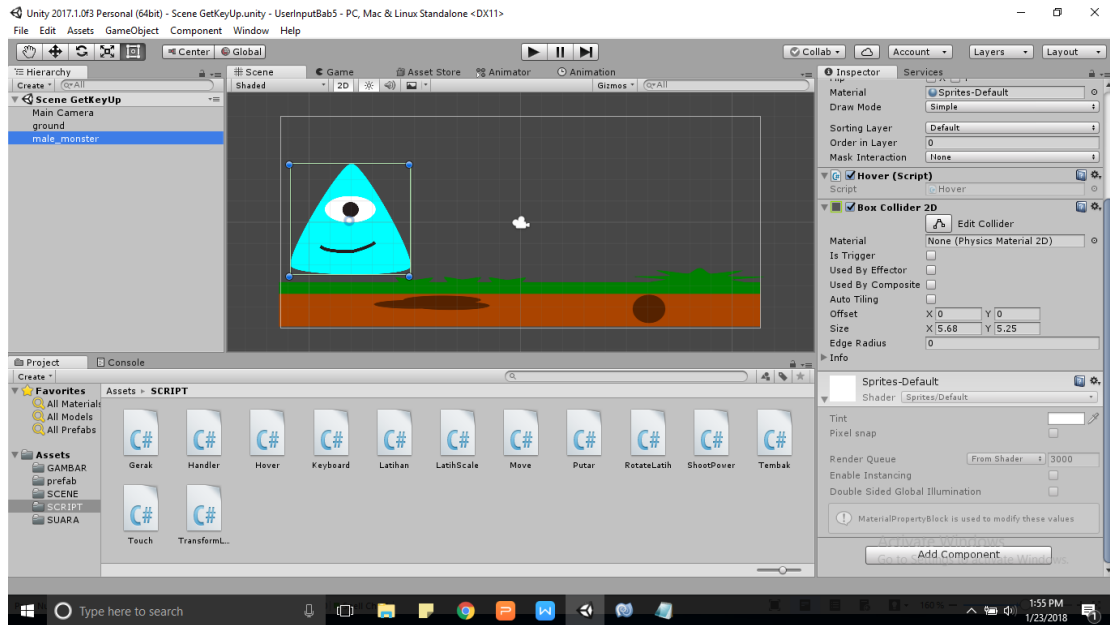
Berikutnya interaksi yang sering dilakukan pemain adalah hover atau menyapu gameobject dengan mouse. Hover sebenarnya sangat sederhana cukup kita tambahkan script yang memiliki void sebagai berikut:

```
void OnMouseOver()  
{}
```

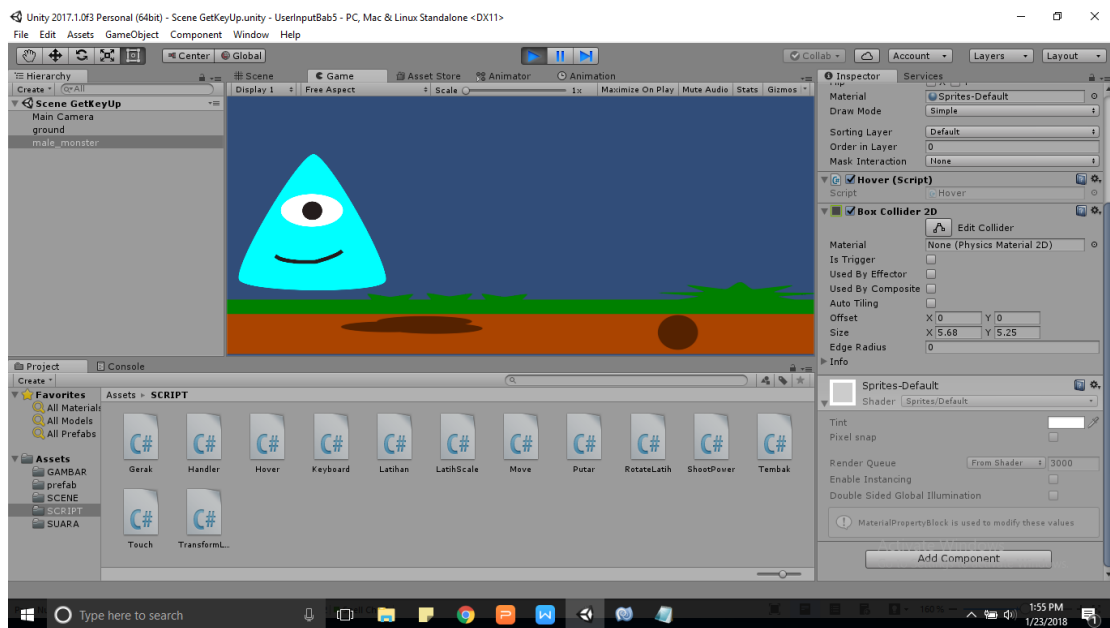
Berikut adalah scriptnya

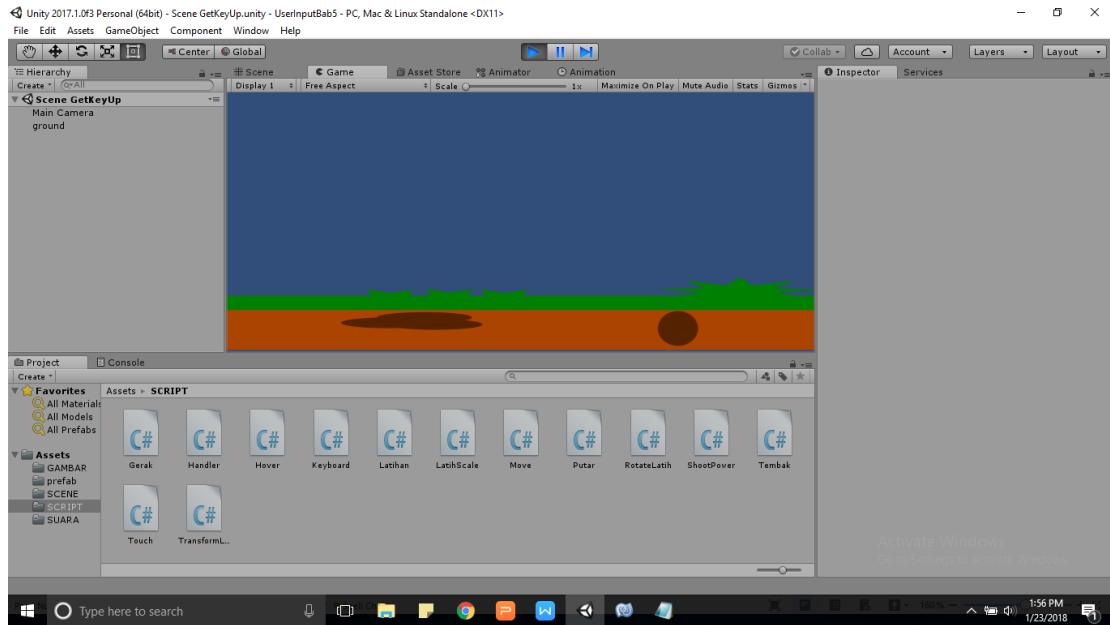
```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Hover : MonoBehaviour {  
  
    // Use this for initialization  
    void Start () {  
  
    }  
  
    // Update is called once per frame  
    void Update () {  
  
    }  
  
    void OnMouseOver(){  
        Destroy (gameObject);  
    }  
  
}
```

Buat gameobject baru yang ditempel komponen script Hover dan juga box collider2d karena script interaksi tidak dapat berfungsi tanpa collider

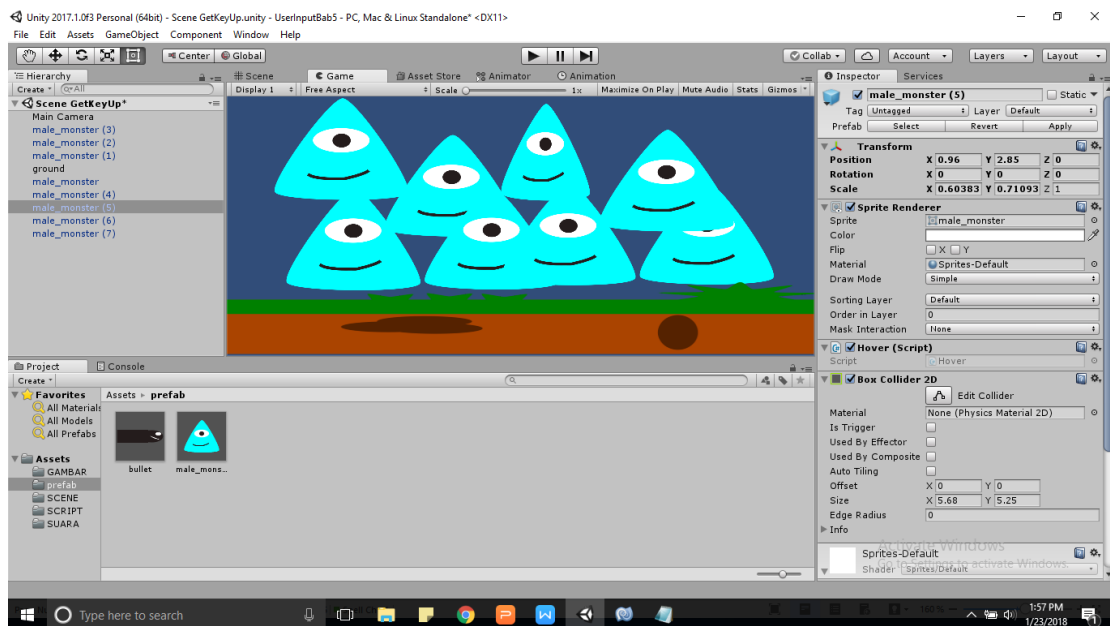


Mainkan lalu hover gameobject tersebut:





Sekarang agar menarik kita jadikan gameobject tersebut sebagai prefab lalu tempatkan disekitar scene



Mainkan dan hilangkan semua monster tersebut dengan cara hover.

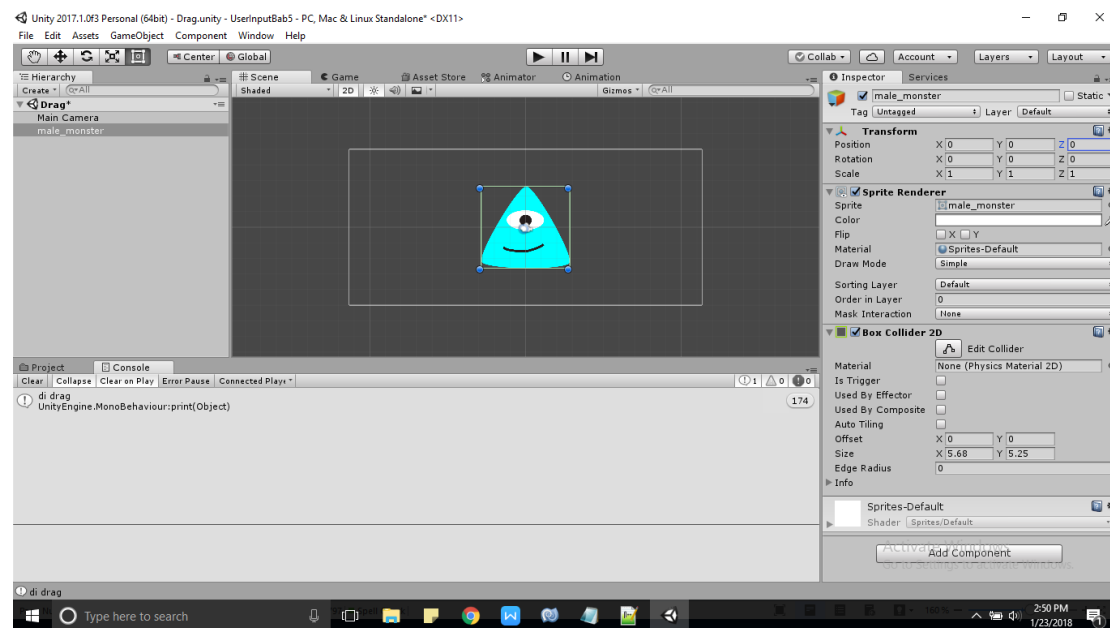
F. Drag And Drop

Berikutnya interaksi yang sering dilakukan pemain adalah drag and drop pertama kita pelajari dulu drag. Drag sebenarnya sangat sederhana cukup kita tambahkan script yang memiliki void sebagai berikut:

```
void OnMouseDown()  
{  
}
```

Namun ada beberapa catatan dan ilmu baru yang dibutuhkan agar fungsi drag ini dapat berfungsi dengan baik. Beberapa contoh dibawah sengaja dibuat salah agar kita dapat memahami perilaku drag.

Pertama buat scene baru lalu tempatkan gameobject anda seperti ini:



Konsep drag and drop sebenarnya sederhana, yaitu selama mouse ditekan maka posisi gameobject mengikuti posisi mouse dimanapun berada. Gambaran scriptnya adalah sebagai berikut:

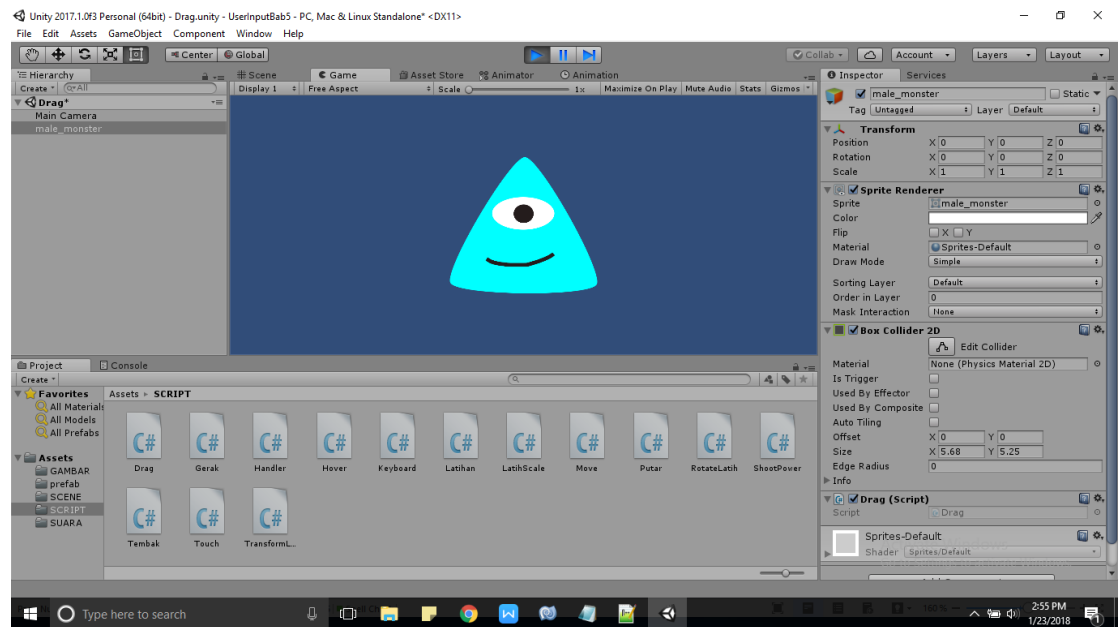
```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Drag : MonoBehaviour{  
  
    // Use this for initialization  
    void Start () {  
  
    }  
  
    public void OnMouseDown ()  
    {  
        transform.position = Input.mousePosition;  
    }  
}
```

```
// Update is called once per frame
void Update () {

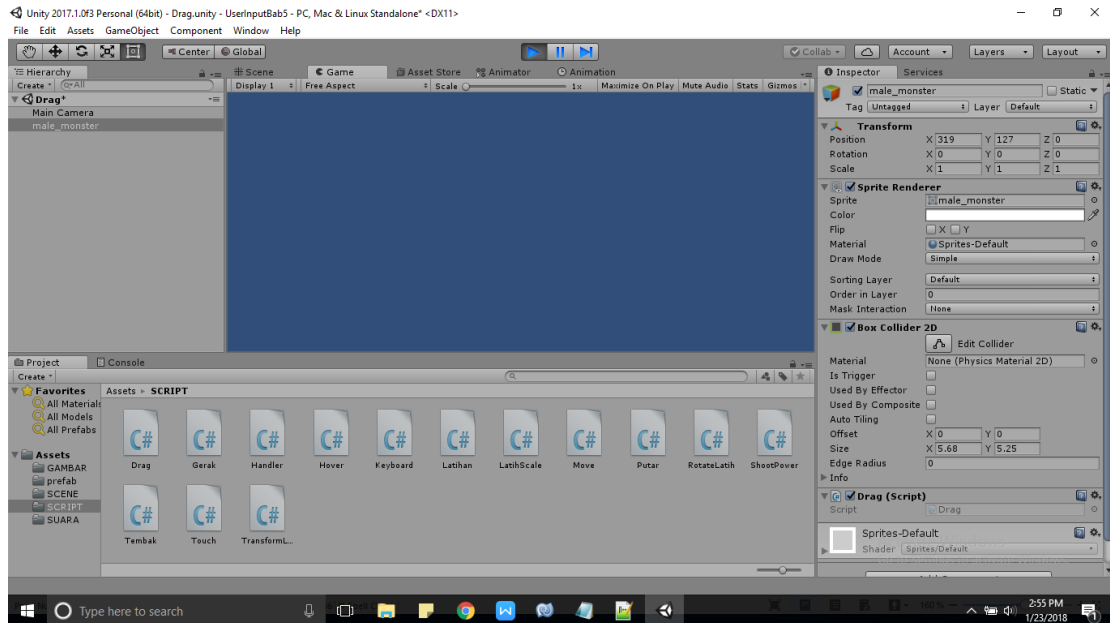
}
}
```

Tempelkan pada gameobject lalu mainkan, sambil drag perhatikan angka pada Transform Position

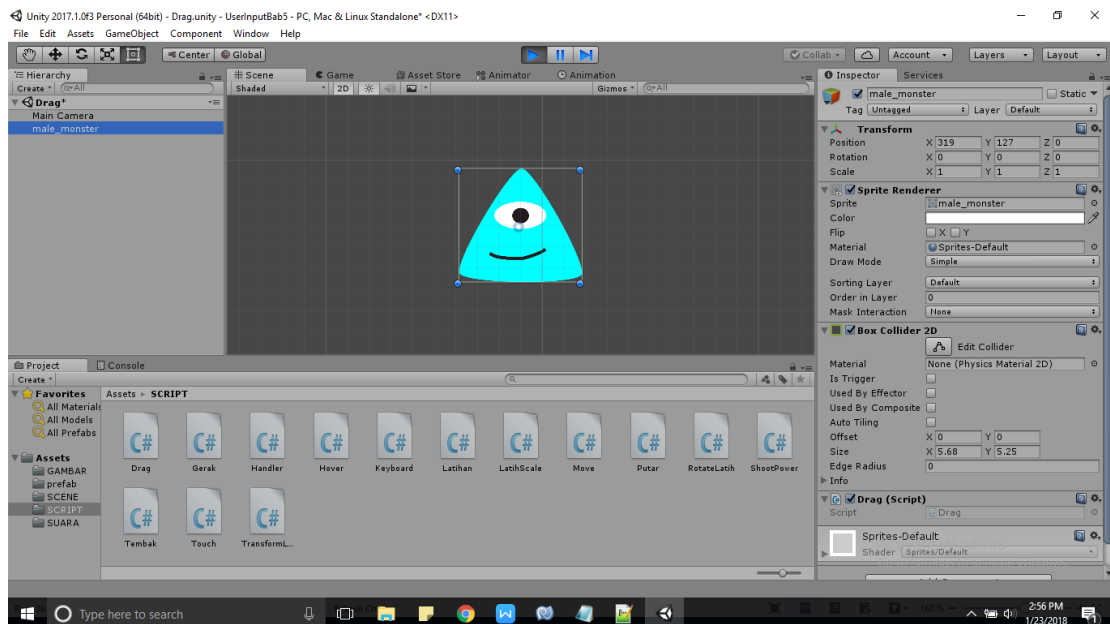
Sebelum di drag, posisinya (x,y,z) adalah (0,0,0)



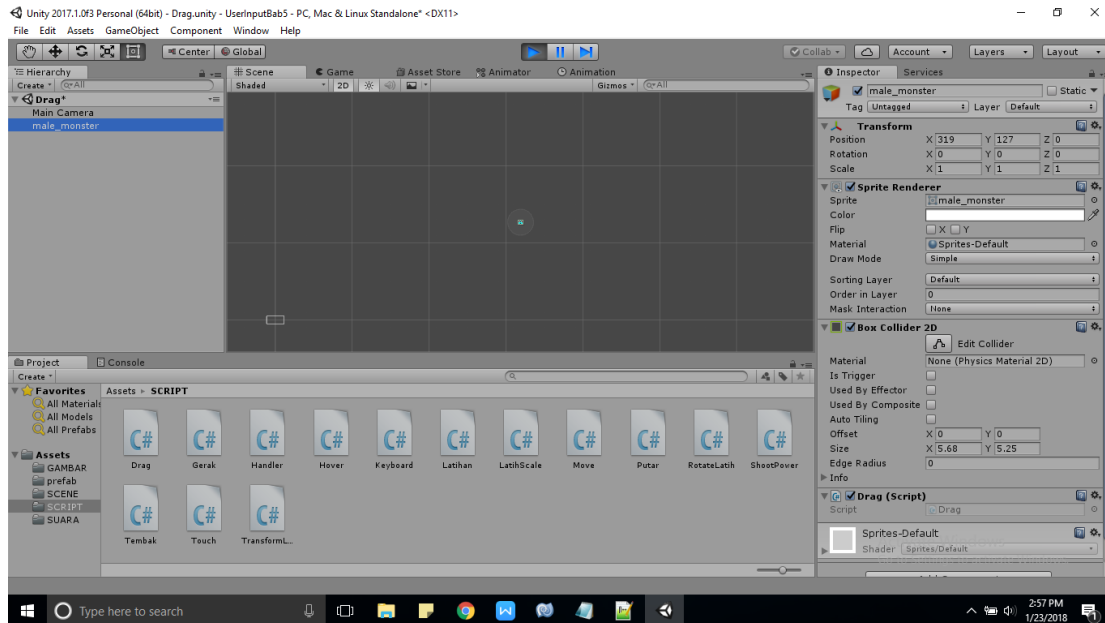
Namun ketika di drag, gameobject kita hilang



Sebenarnya gameobject kita bukan hilang tp berpindah ke posisi yang sangat jauh dan tidak terjangkau oleh ruang main camera. Untuk melihat dimana gameobject kita berada buka tab scene dan klik dua kali pada gameobject kita



Muncul bukan, dan kalau kita coba petakan maka kita bisa melihat bahwa ruang main camera dan gameobject berada dalam posisi yang sangat berjauhan



Mengapa hal ini bisa terjadi? Hal ini terjadi karena antara gameobject 2d dengan mouse memiliki koordinat yang berbeda, mouse memiliki koordinat 3 dimensi sedangkan gameobject kita koordinatnya adalah 2 dimensi. Maka langkah berikutnya kita harus menyesuaikan posisi gameobject 2d kita menjadi format 3 dimensi sebagai berikut:

```

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

public class Drag : MonoBehaviour{

    // Use this for initialization
    void Start () {

    }

    public void OnMouseDown ()
    {
        transform.position = new Vector3 (Input.mousePosition.x, Input.mousePosition.y,0f);
    }

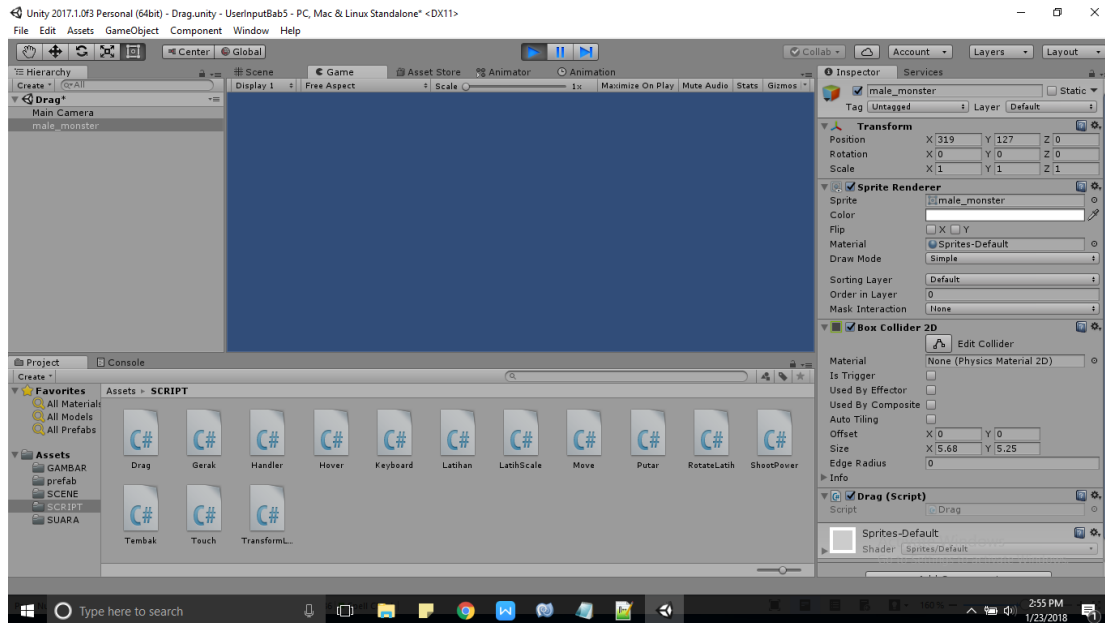
    // Update is called once per frame
    void Update () {

    }

}

```

Silahkan mainkan lalu coba drag



Ternyata masih hilang, hal ini dikarenakan format koordinat mouse tidak mampu mengenali format koordinat camera sehingga yang harus kita lakukan adalah mengkonversi koordinat mouse ke koordinat camera dengan syntax `Camera.main.ScreenToWorldPoint`. Perlu diketahui bahwa antara koordinat mouse dengan koordinat camera memiliki selisih 10 satuan terhadap sumbu z, sehingga untuk mengkonversi sekaligus mengakali selisih sumbu z antara camera dan mouse maka scripturnya adalah sebagai berikut:

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

public class Drag : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

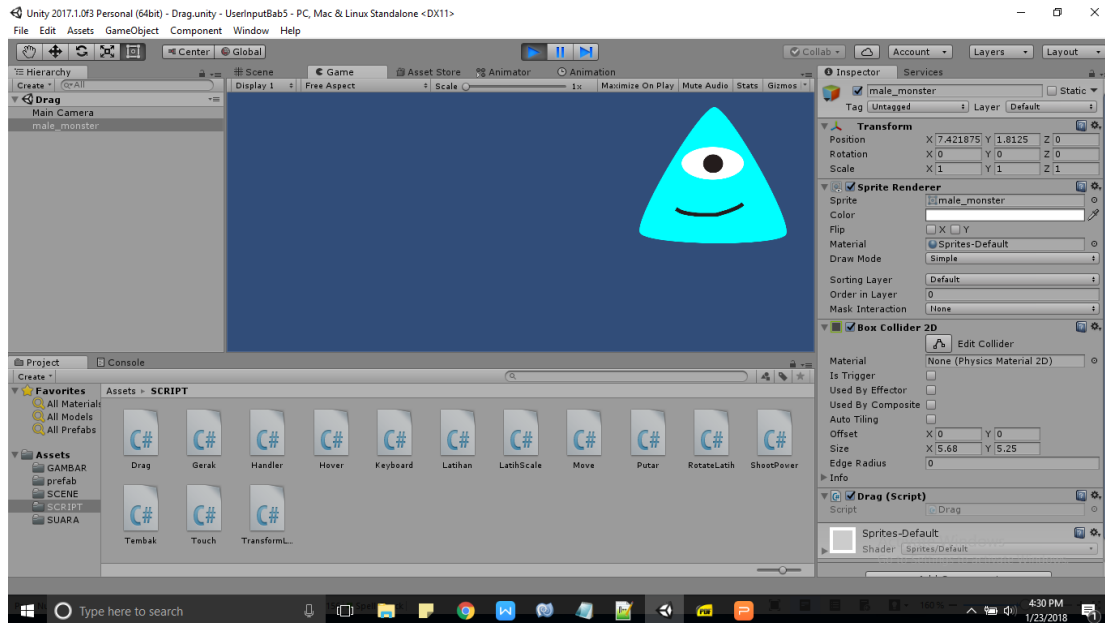
    public void OnMouseDown ()
    {
        transform.position = Camera.main.ScreenToWorldPoint(new Vector3 (Input.mousePosition.x, Input.mousePosition.y, 10f));
    }

    // Update is called once per frame
    void Update () {

    }

}
```

Silahkan mainkan, dan selamat melakukan drag



Untuk drop kita hanya perlu menambahkan void sebagai berikut:

```
public void OnMouseUp () {}
```

Berikut adalah scriptnya

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

public class Drag : MonoBehaviour {

    // Use this for initialization
    void Start () {

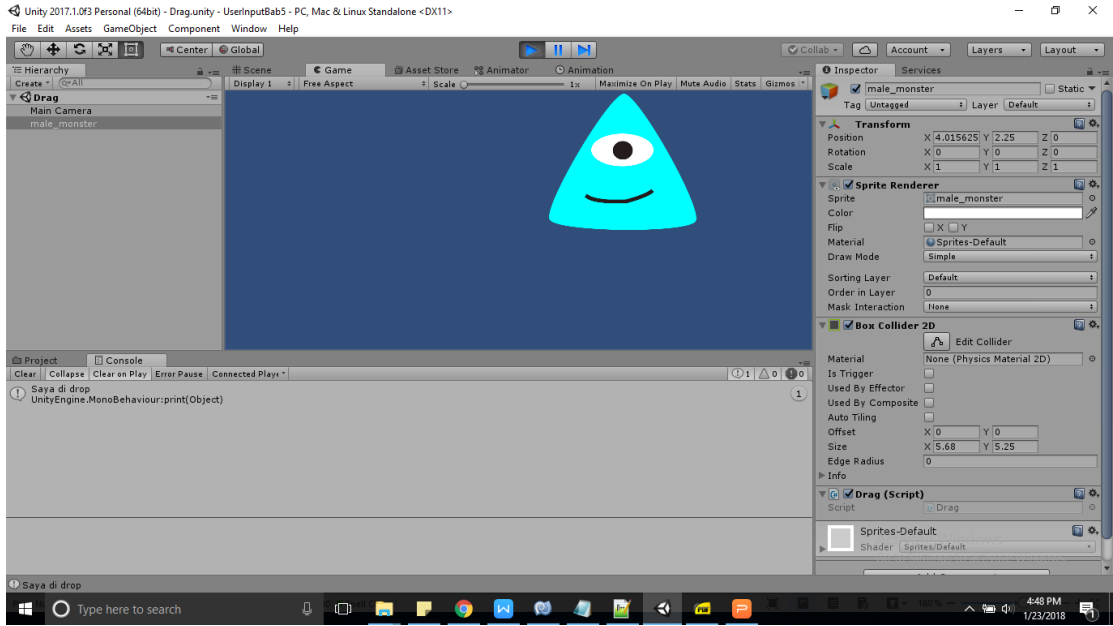
    }

    public void OnMouseDown ()
    {
        transform.position = Camera.main.ScreenToWorldPoint(new Vector3 (Input.mousePosition.x, Input.mousePosition.y,10f));
    }

    public void OnMouseUp ()
    {
        print ("Saya di drop");
    }

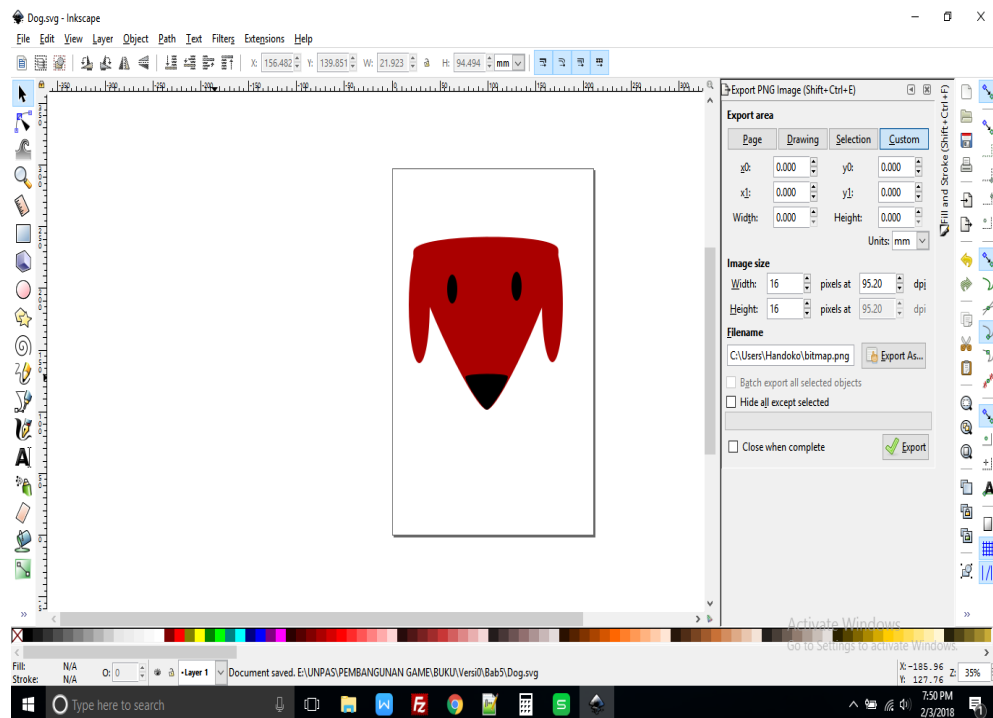
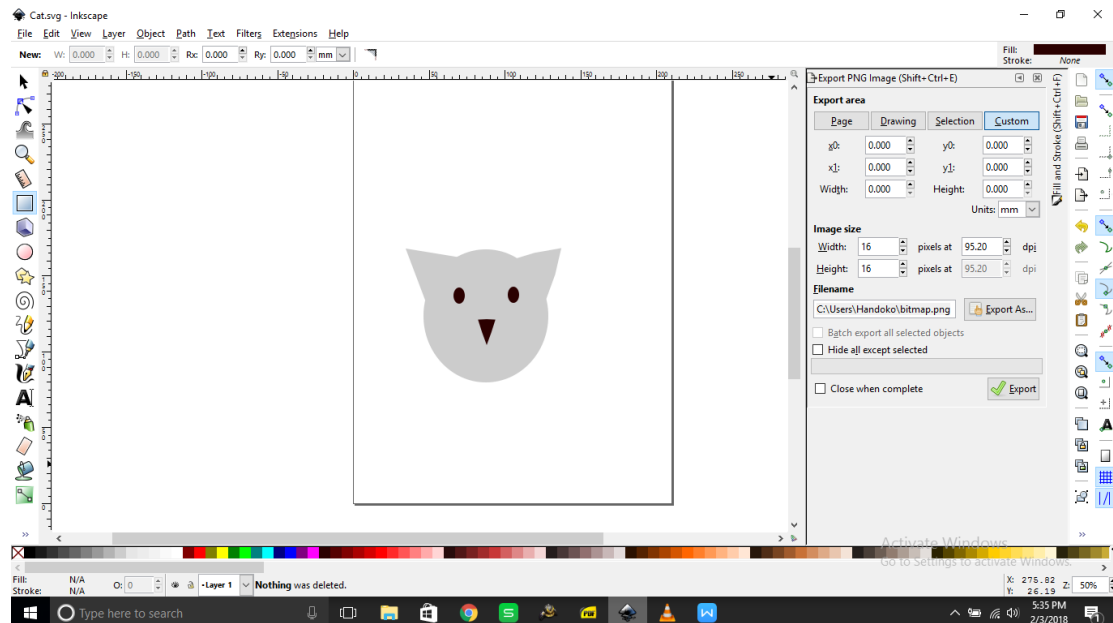
}
```

Silahkan dimainkan maka setelah melakukan drag and drop maka akan muncul tulisan sebagai berikut:

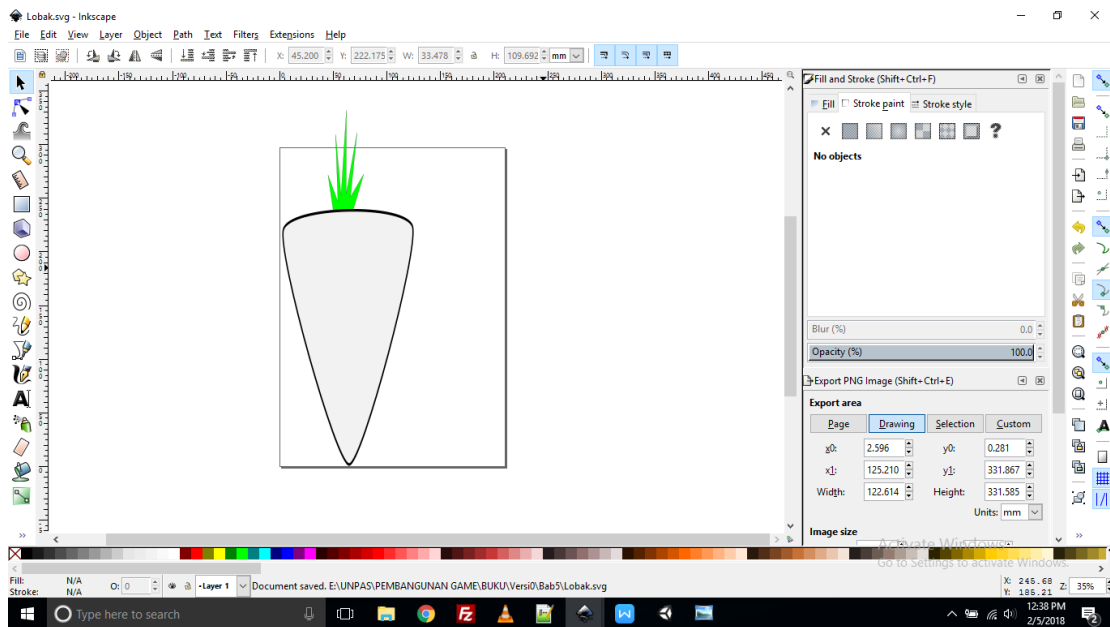
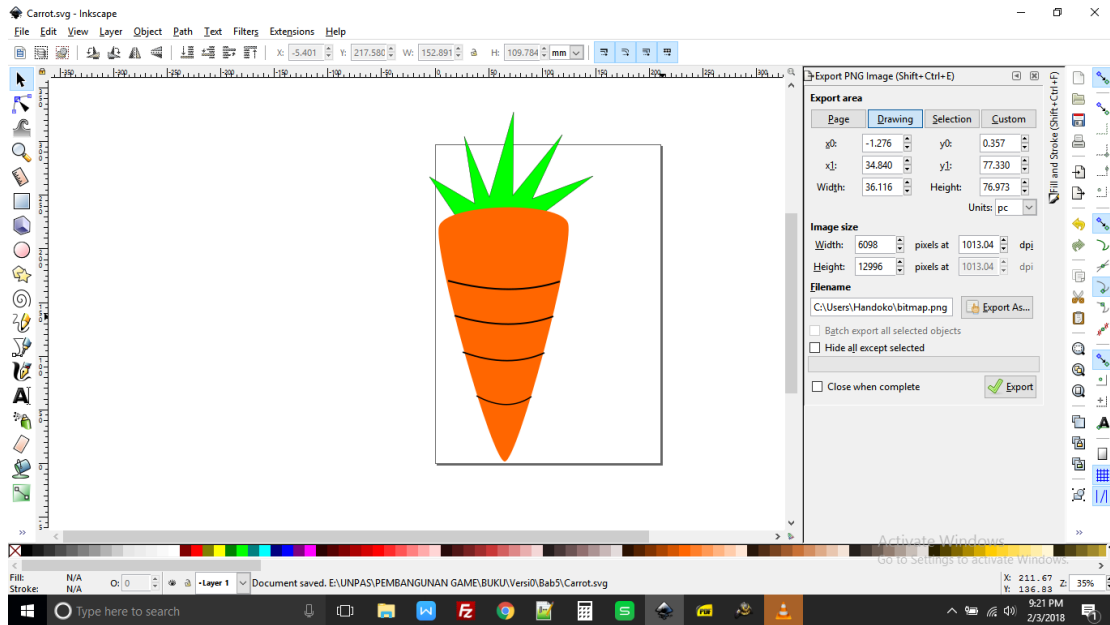


G. Drag And Place

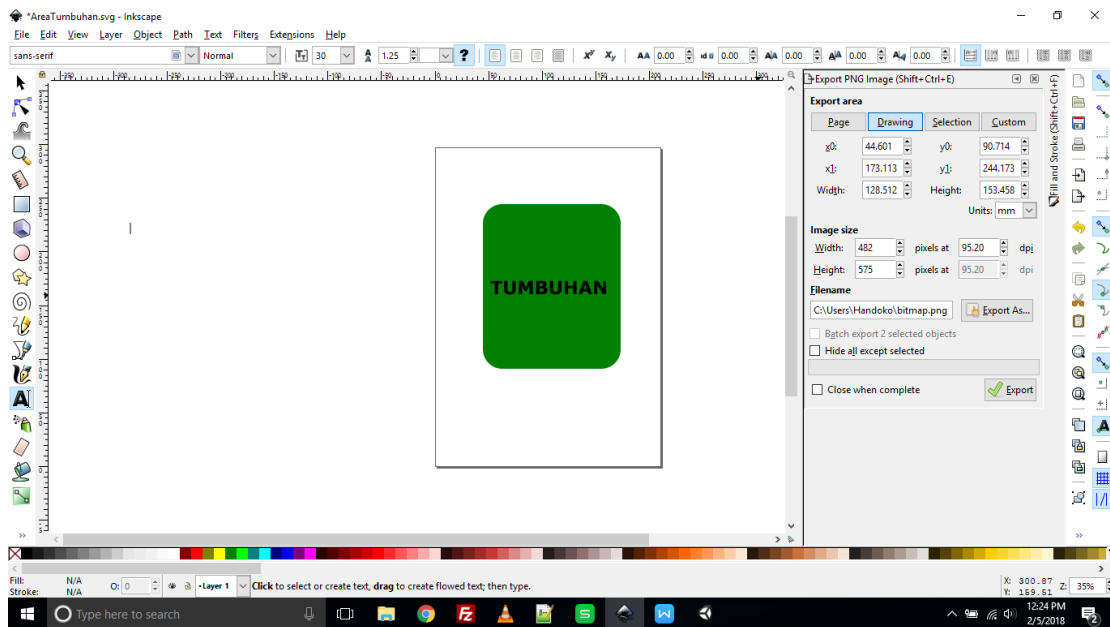
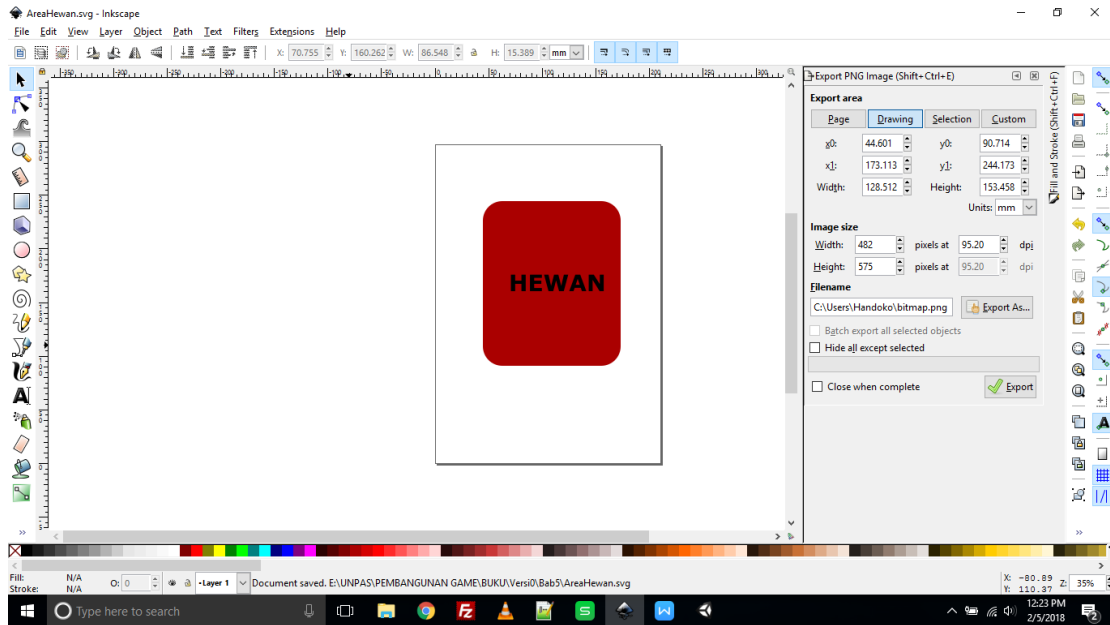
Berikutnya interaksi yang sering dilakukan pemain adalah drag and place. Drag and place merupakan pengembangan dari teknik Drag and Drop, namun bedanya pada Drag and Place gameobject yang kita drag haruslah kita drop pada tempat(place) yang valid. Jika tidak valid maka drop gameobject tersebut dibatalkan dan gameobject tersebut kembali ke posisi awal. Validasi gameobject yang di drag terhadap tempatnya adalah menggunakan tag. Tag merupakan pengkategorian gameobject, misal kucing dan anjing merupakan kategori(tag) hewan dan wortel serta tomat merupakan kategori(tag) tumbuhan. Mari kita mencoba dulu mendalami tag di unity. Pertama buatlah gambar kucing dan anjing di inkscape sebagai berikut:



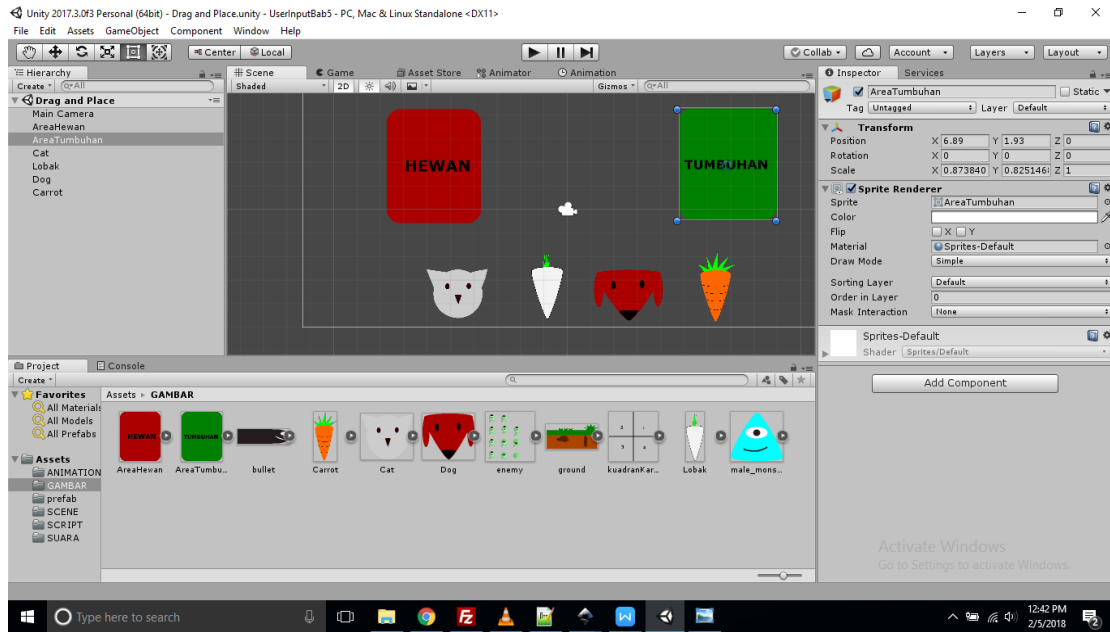
- Juga wortel dan lobak



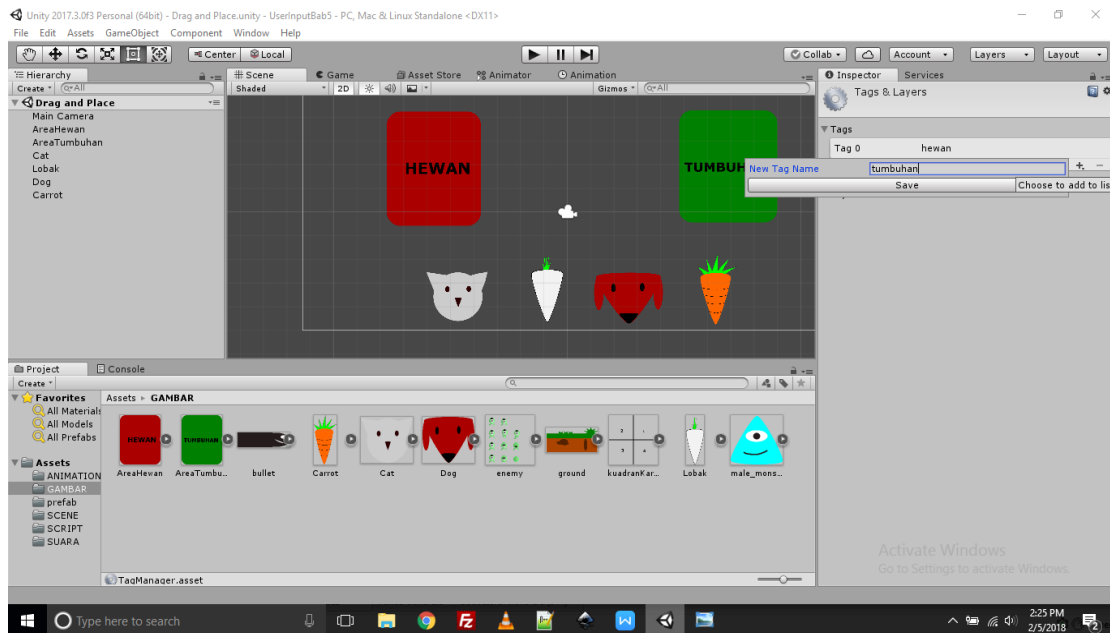
Sekaligus kita siapkan juga tempat nanti kita menempatkan gambar kita saat nanti di drag and place di inkscape



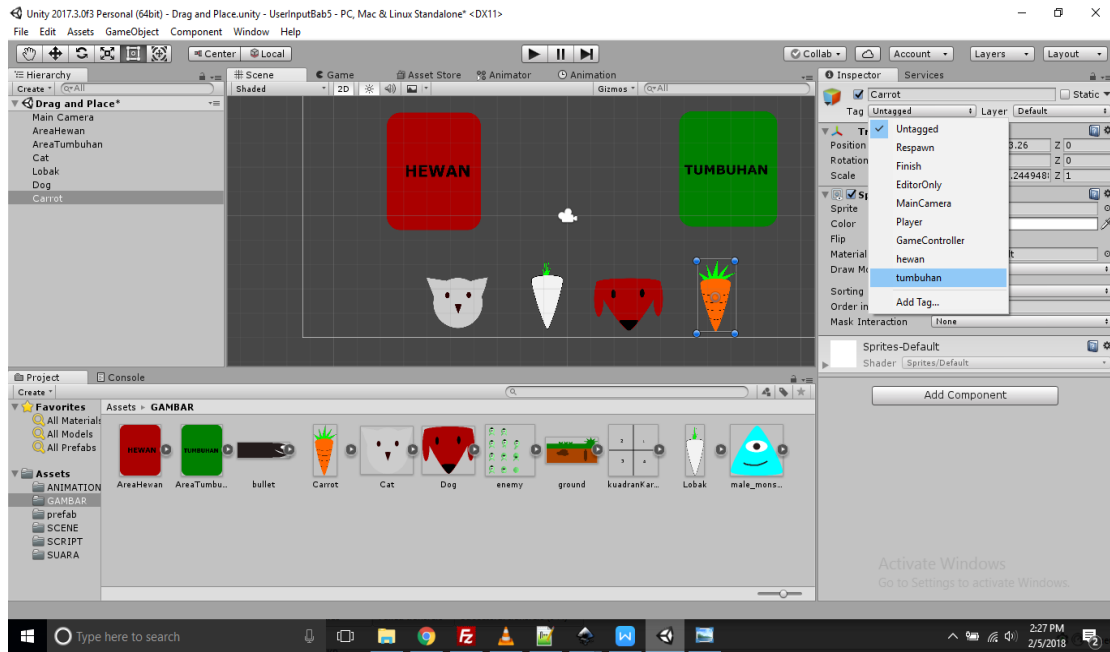
Jika sudah buat scenenya sebagai berikut:



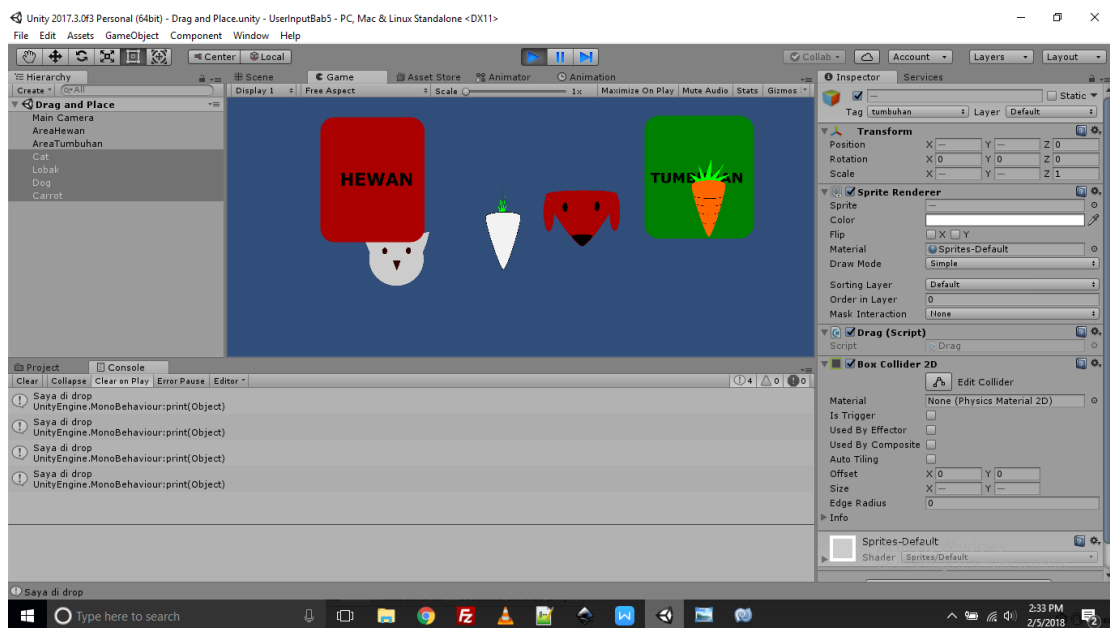
Baik sekarang kita tambahkan tag yaitu hewan dan tumbuhan, seperti ini:



Berilah area hewan, kucing dan anjing tag hewan serta area tumbuhan, wortel dan lobak sebagai tumbuhan, seperti ini:

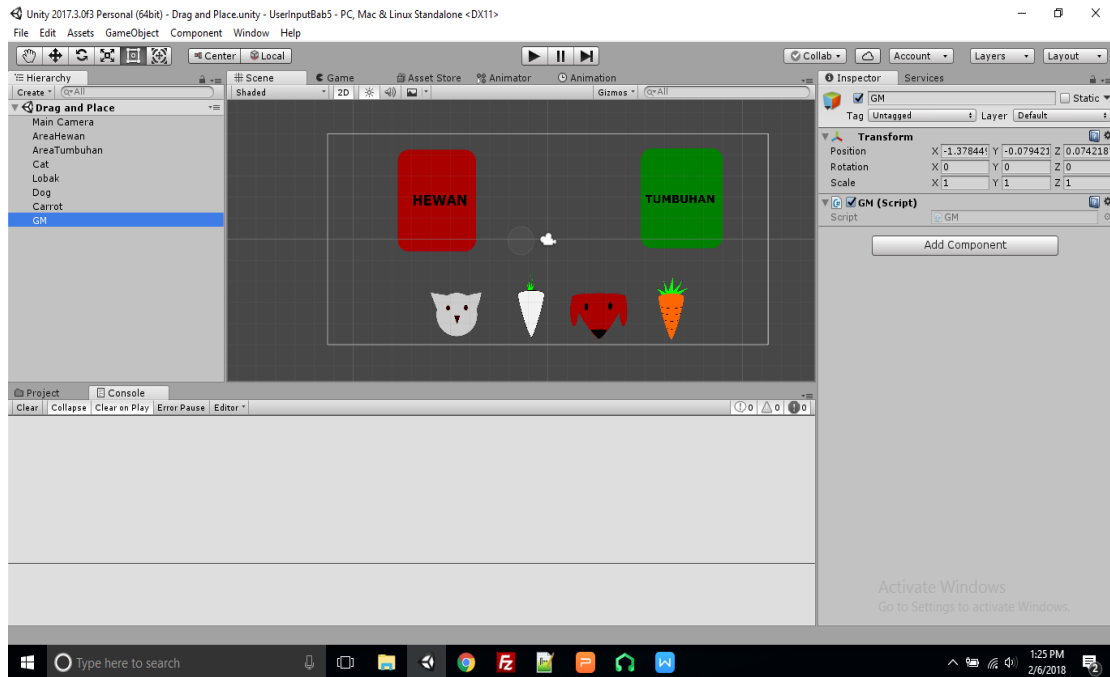


Selanjutnya tambahkan script drag and drop yang pernah kita buat kepada anjing, kucing, wortel, serta lobak sehingga kini keempat gameobject tersebut dapat didrag.



H. Game Manager (GM)

Pertanyaannya adalah bagaimana agar kita dapat menentukan bahwa object yang kita drag ditempatkan di kotak benar yang memiliki tag sama. Disinilah kita perlu wasit, dengan asumsi bahwa baik area placed dan gameobject drag adalah pemain maka perlu wasit yang menentukan bahwa gameobject di drag and place di kotak yang benar atau salah. Wasit tersebut bernama Game Manager (GM). Untuk membuat GM buat gameobject baru lalu juga buat script baru yang bernama GM.cs lalu jadikan komponen script GM sebagai berikut:



Koding di GM.cs adalah sebagai berikut:

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

public class GM : MonoBehaviour {

    RaycastHit2D[] hits;
    GameObject pair1,pair2,tmp;
    int i;
    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {
        hits = Physics2D.RaycastAll(Camera.main.ScreenToWorldPoint(Input.mousePosition),Vector2.zero);
        if (hits.Length > 0)
            tmp = hits [hits.Length - 1].transform.gameObject;
    }

    public void setPair1(GameObject go){
        pair1 = go;
    }

    public void setPair2(){
        pair2 = tmp;
    }

    public bool valid(){
        bool ntv = false;
        if (pair1.tag == pair2.tag)
            ntv = true;
        return ntv;
    }
}
```

Buat juga script DragPlace yang menggantikan script Drag pada gameobject drag di scene. Script DragPlace adalah sebagai berikut:

```

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

public class DragPlace : MonoBehaviour{

    Vector3 posisiAwal;

    // Use this for initialization
    void Start () {
        posisiAwal = transform.position;
    }

    public void OnMouseDown ()
    {
        GameObject.Find ("GM").GetComponent<GM> ().setPair1 (gameObject);
    }

    public void OnMouseDrag ()
    {
        transform.position = Camera.main.ScreenToWorldPoint(new Vector3 (Input.mousePosition.x, Input.mousePosition.y,10f));
    }

    public void OnMouseUp ()
    {
        transform.position = posisiAwal;
        GameObject.Find ("GM").GetComponent<GM> ().setPair2 ();
        if (GameObject.Find ("GM").GetComponent<GM> ().valid ())
            Destroy (gameObject);
    }

}

```

Ketika dimainkan lalu kita mendrag kucing ke kotak yang benar maka kucing menghilang, hal ini bisa dikembangkan dengan score dsb.

