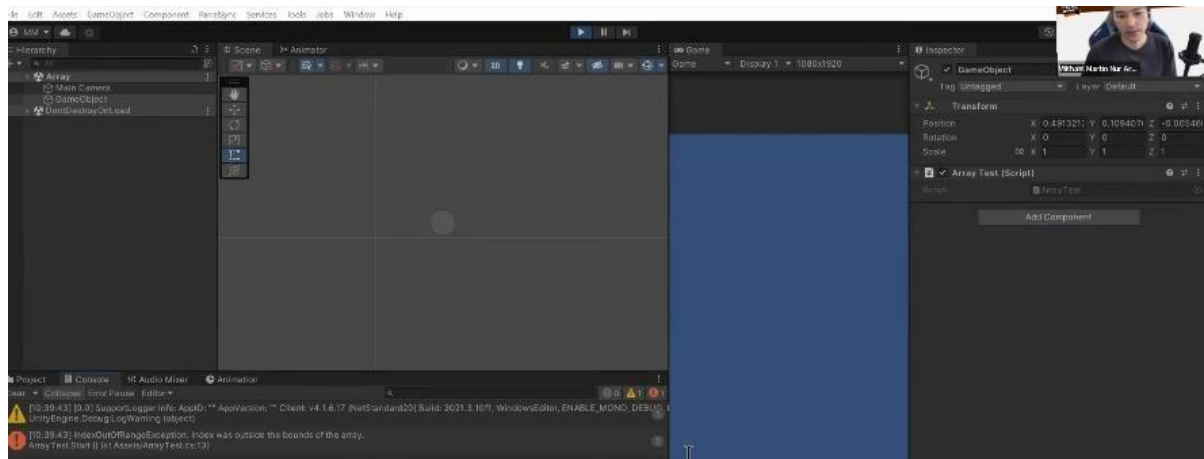


Writing & Presentation Assignment Week 9

UNITY PROJECT OPTIMIZATION

Review materi tentang rock-paper-scissors. Selain itu kak mikha, menjelaskan cara membuat array di proyek rpc dibuat di scene kosong Bernama array.



Buat kodingan seperti berikut :Kondisi array saat awake :

```
Int[] intArray = new int [] {  
    1,2,3 //index dimulai dari 0  
};
```

Index of range = kita mengakses array dari array yang ada.

Bagusnya harus ngecek lebih dahulu :

```
Int[] intArray = new int [] {  
    1,2,3 //!index dimulai dari 0  
};  
  
Int selectedIndex=3;  
Void Start() {  
  
    //!kalau ngga ada = -  
    1  
  
    Debug.log(intArray.L  
    ength);Int  
  
    selectedIndex = - 1;  
  
    //!kalau ada ngambil nilai, untuk menghindari problem unity
```

```

//If(selectedIndex < intArray.Length && selectedIndex==0) //kalau ngga >= 0
    akan errorselectedNum = intArray[selectedIndex];

Debug.log(intArray.Length);
}

```

Jadi C# untuk handle exceptions punya kelas dan diturunkan, seperti kode berikut

```
Public sealed class IndexOutOfRangeException : SystemException
```

Untuk mencari problem bisa menggunakan kode berikut

```

Catch (System.Exception e) {

    Debug.Log (e)

    throw;

}

Debug.Log(selectedNum);

```

Jika ada error lain bisa buka dokumentasi Microsoft, karena disana banyak script

Try catch bisa digunakan untuk mengecek ada tidaknya suatu nilai, tapi bisa menambahkan nilai default

```

Try {

    selectedNum = intArray[selectedIndex];

}

Catch (System.Exception e) {

    Debug.Log (e)

    selectedNum=0; throw;

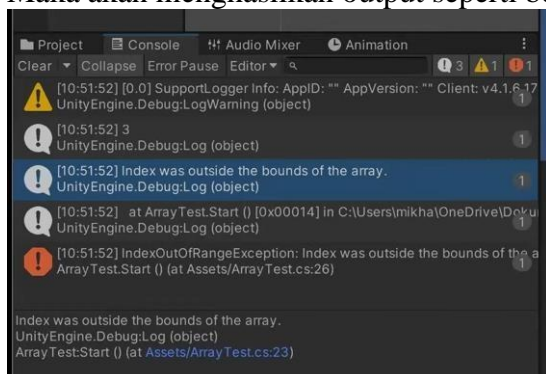
}

Debug.Log(selectedNum);

```

Fungsi try-catch = untuk mengecek semua kode, menghindarkan program error, dan hal – hal yang tak terhindarkan, missal internet terputus, maka try-catch ini bisa manage tadi.

Maka akan menghasilkan output seperti berikut :



Message nya itu yang error merah, Unity biasanya format Ketika error dimulai dari **jenis nya apa : pesannya. Letak errornya di script mana (nama script yang error)**

Dan Ketika di double klik akan teleport ke script yang error StackTrace nya itu yang ArrayTest.Start()

//agar tidak error, bisa dihapus thrownya, karena throw untuk melemparkan eror

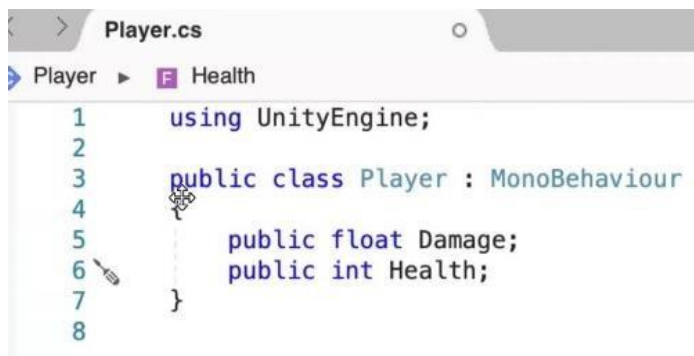
```
Try {  
    selectedNum = intArray[selectedIndex];  
}  
Catch (System.Exception e) {  
    Debug.Log (e.Message);  
    Debug.Log(e.StackTrace);  
    selectedNum=0;  
}
```

Debug.Log(selectedNum); //maka error akan menghilang

Bisa juga pakai **Debug.LogError**, maka akan terlihat merah di console unity

SCRIPTABLEOBJECT

Review materi tentang projek dan membahas **ScriptableObject**. **ScriptableObject** sebenarnya berupa asset tapi dimulai dari fungsinya sebagai data container (bukan **MonoBehaviour**) bisa menyimpan data banyak.



Kalau di kelas **MonoBehaviour** : Data in game, bikin variable di kelas, disimpan di GameObject dan baru valid (Instantiate).

Perbedaannya **ScriptableObject** dengan **MonoBehaviour**:

1. Kalau **ScriptableObject** lebih ke arah asset bukan GameObject dalam scene
2. Dia tidak bisa di **Instantiate di scene**, tapi **ada di luar projek**.
3. Bayangkan **game** merupakan **level projek** yang didalamnya banyak **scene** (level), hidup di projek. Selama projek menyala, dia hidup. Kalau scene di close, si asset masih ada (bernilai). Dia bisa mengoper data antar scene.
4. Hanya di unity editor, misal saat play mode, developer bisa mengubah – ubah nilai **ScriptableObject** asset, Ketika selesai, maka perubahan akan tersimpan. Kecuali, Ketika di build maka posisi Kembali ke awal.

Beberapa alasan **ScriptableObject** dibutuhkan :

1. Memudahkan saving & storing data Ketika editor session
2. Saving data sebagai asset untuk digunakan bersifat runtime (hidup di memori bisa diakses semua / *middleman*)
3. Mengoptimasi data dengan mengolah duplikasi data penggunaan memori
4. Sebagai enum

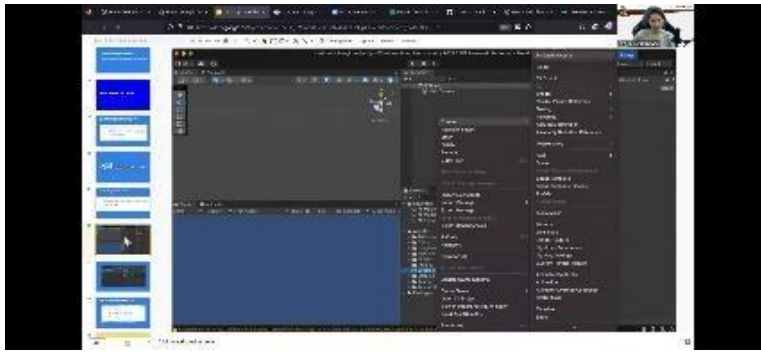
Cara membuat **ScriptableObject** :



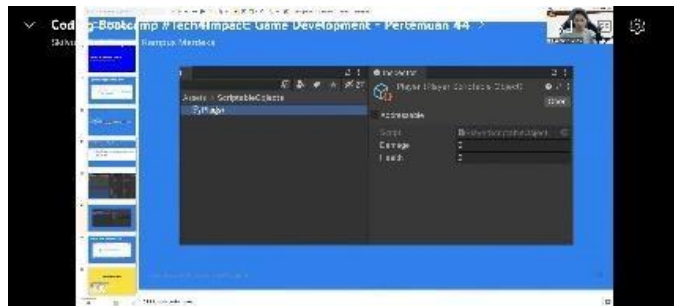
Pertama, buat asset dahulu dan tambahkan **CreateAssetMenu** diatasnya



Nanti dengan adanya ini, akan muncul menu ini agar bisa dibuat



Dan jadi asset yang ada di proyek folder berupa file. Maka si data tadi akan ada di inspector



Dalam **ScriptableObject** tetap butuh **MonoBehaviour**, tapi disini kita punya semacam data player yang diambil dari asset (tidak dimasukkan ke inspector untuk gameobject player).

Fungsinya Ketika ingin dirubah, missal enemy yang berbeda, bisa jadi menggunakan script **behaviour** sama tapi beda di state saja (size, speed, damage) maka harus menggunakan **ScriptableObject**.