Reflection

**Oyun içi gereksinimler;**

1 – Level tasarımı

2 – Başlangıçta minimum 100 Level

3 – Animasyonlar

4 – Her levele uygun zorluk derecesi

5 – Yandığında video izleyerek devam edebilme özelliği(1 kereye mahsus)

6 – Market ile rasgele karakter alınması.

**Yapılacaklar;**

1 – Arayüz tasarımı, arayüzde neler olması gerekiyor.

2 – Değişkenleri global yapılacak.

**Awake;**

Awake çalıştığında activeLevelid si ile bölümün geçilip geçilmediğine bakıyor.

Duruma göre levelin button enablını true yapıyor ve üzerindeki çarpıyı kaldırıyor.

**Levels Class:**

[System.Serializable] public class Levels

{

public Modes mode; = Oyunun mode tipini beliriyor.

public Material skyboxMaterial; = Background ayarlıyor.

public GameObject levelX; = Bölüm üzerindeki X içerini kaldırmak için, her zamana bir sonraki levelinX ini alıyor.

public Button enableButton; = Level tıklanma için. Her zaman bir sonraki levelin buttonunu alıyor.

public float characterSpeed; = Karakterin hızı

public float enemySpeed; = Gelen objelerin hızı

public float gameTime; = Oyun süresi

public int maxBallCount; = Toplanacak toplam obje

public string levelId; = Level id (1 den başlıyor)

}

Reflection

public void **StartGame**(int \_id)

{

nextEnableButton = levels[\_id-1].enableButton;

activeLevel = \_id;

freeModeFlag = false;

DeleteSpheres();

StopAllCoroutines();

startPanel.SetActive(false);

maxBallCount = levels[\_id-1].maxBallCount;

nextLevelX = levels[\_id-1].levelX;

stagesButton.SetActive(false);

Main.speed = levels[\_id - 1].characterSpeed;

enemySpeed = levels[\_id - 1].enemySpeed;

StartCoroutine(CreateSphere(levels[\_id-1].mode));

}

**StartGame** fonksiyonu bütün level metodlarının içinde mevcut

\_**id** olarak aldığı parametre (1 den başlıyor) active level belirlemek için kullanılıyor