

BLM0364 Oyun Programlama

2022-2023 Güz

Vize Projesi

İki kişilik ekipler halinde Unity oyun motorunu kullanarak aşağıdaki kriterleri karşılayacak şekilde projenizi yapınız. Tüm scriptlerinizi C# dili ile kodlayınız. Projenizi bir Github reposuna yükleyiniz (her ekip üyesi ayrı ayrı yüklemelidir). Github reponuzun readme sayfasına yine aşağıda belirtildiği gibi gerekli açıklamaları ve raporu ekleyiniz. En sonunda oyununuzu WebGL platformu için derleyiniz ve dosyaları itch.io veya simmer.io gibi bir platforma yükleyiniz. Projeniz açık kaynak olarak Github'da yer almalı fakat oynanabilir sürümü tarayıcıdan çalıştırılabildiği sürece herhangi bir platformda olabilir.

Projenin Gereklikleri

- Oyununuzda girdiler için sadece Input.GetAxis() (veya Input.GetAxisRaw()) kullanınız. Her bir girdi **+10**, GetButtonUp veya GetKey gibi diğer girdi alma yolları **-5** olarak puanlandırılır. (Aynı işlevi gerçekleştiren tuşlar 1 kez puanlandırılır.)
- Oyununuz fizik motorunu kullanıyor olmak zorunda değildir. Fakat kullanılan ve doğru/planlandığı gibi çalışan her OnCollisionXXX ve OnTriggerXXX fonksiyonları **+10** olarak puanlandırılır. (Aynı işlevi gerçekleştiren fonksiyonlar 1 kez puanlandırılır.)
- Oyununuz tek bir oynanış sahnesi içermeli ve direkt o oynanış sahnesinden başlamalıdır. Ana menü veya geçiş sahnesi koymayınız. Oynanış sahnesinde başlangıçta kapalı olan bir durdurma menüsü (pause menu) olmalıdır. Klavyenin "Esc" tuşuna basıldığında açılmalı ve tekrar basıldığında kapanmalıdır. Esc tuşuna basılı tutmak menüyü açıp kapamamalıdır. Sorunsuz açılıp kapanan menü **+20** olarak puanlandırılır.
- Durdurma menüsünün sağ üst kısmında oyununuzun adı, sol alt kısmında ise alt alta "Devam et" ve "Yeniden başlat" butonları bulunmalıdır. Menü açıldığında oyun durmalı (veya durma noktasına gelecek kadar yavaşlatılmalı), menü kapandığında (Esc veya "Devam et" butonu; ikisi de implement edilmelidir) oyun aynı hızında devam etmelidir. Sorunsuz çalışan "Devam et" butonu **+10**, sorunsuz çalışan "Yeniden başlat" butonu **+10** olarak puanlandırılır.

*Not: Yeniden başlatma için scriptinizin en başına
using UnityEngine.SceneManagement;
ve buton aksiyonuna*

*SceneManager.LoadScene(SceneManager.GetActiveScene().name);
koyabilirsiniz.*

Projenin Raporlanması

Github reponuzun readme sayfasını rapor olarak kullanınız. Raporunuz bireysel olarak hazırlanmalı ve şu başlıkları içermeli:

- Oynanış ve -varsa- oyunun hikayesi
- En az 1 oyun içi görsel
- Tuşlar
- Oyunda sizin geliştirdiğiniz kısımlar ve kodladığınız scriptler
- Oyunda ekip arkadaşınızın geliştirdiği kısımlar ve kodladığı scriptler

Gönderim

Ekampus üzerinden **sadece** Github reponuzun adresini gönderiniz. Her ekip üyesi bireysel olarak gönderim yapmalıdır. Hatalı repo bağlantısı olarak, Github profil linki olarak veya yazı/not eklenerek bağlantısı gönderilen projeler değerlendirme dışı bırakılır.

Örnek repo adresi: <https://github.com/kullaniciadiniz/oyunprojeniz>

Dikkat

Projenizde kaynak göstermek koşuluyla harici varlıklar (2D-3D grafikler, materyaller, shaderlar, sesler) kullanabilirsiniz. Harici hiçbir scripti projenizde tutmamalısınız. Bulduğunuz scriptlerin satırlarını kopyalayıp veya karıştırmanız, değişken/fonksiyon isimlerini refactor etmeniz intihal olmasını engellemez. **Harici script** veya **intihal** içeren projeler **değerlendirme dışı** bırakılır.

Not: Projenizi, eğitim videolarından faydalanarak kodlamayınız. Aynı videoyu izleyip kod yazan tanımadığınız kişiler daha önce internet üzerinde bunları yayınlamış olabilir. Bu da sizin kodunuzun intihal olarak işaretlenmesine sebep olacaktır.

Aynı oynanış mekanizmalarına sahip projelerin her birinden ortak mekanizmalı proje sayısı x 10 kadar puan kesilir. Örn; engel atlama oyunu yapan 3 grup 70 üzerinden değerlendirilir.

Son teslim tarihinden sonra Github reposunda değişiklik yapılan projeler değerlendirme dışı kalır. Github değişikliklerinizi gizleyemeyeceğinizi ve geri alamayacağınızı unutmayınız.

Başarılar...