

CHALLENGE FOR UNITY DEVELOPER



GAMEGURU



Merhaba, challenge bağımsız iki ayrı projeden oluşmaktadır.

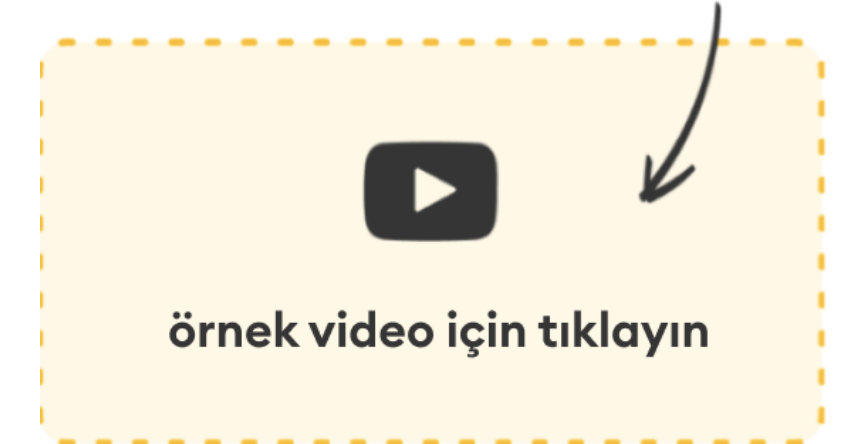
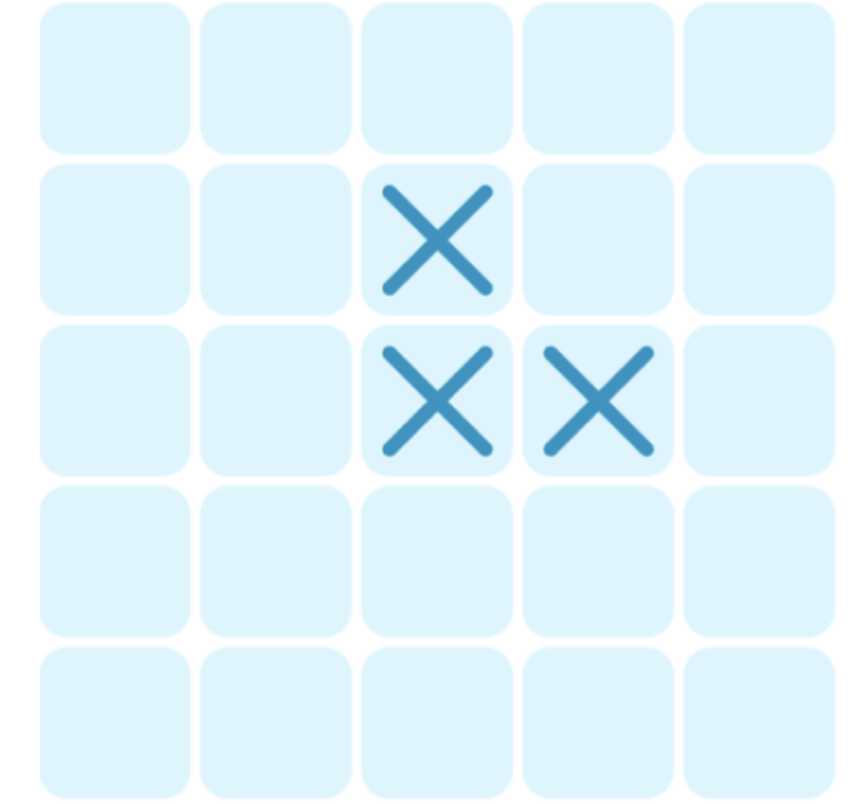
Her iki proje için de:

- Unity **2020.3.x** veya **2021.3.x** versiyonlarından birini kullanmalı,
- Projeyi en baştan itibaren git ile versiyonlamalı ve düzenli commitler gerçekleştirmeli,
- Projeni GitHub, Bitbucket veya benzeri bir platform'a public olarak yükleyerek linkini paylaşmalısın
- (BONUS) olarak işaretlenmiş maddeleri hayata geçirmek senin tercihine bağlıdır
- SOLID prensiplerine bağlı kalmalı
- Herhangi bir Dependency Injection framework'ü kullanılmalı (BONUS)
- Kod okunaklı ve tercih edeceğin bir coding convention'a bağlı olmalıdır.

PROJE 1



1. Editör’de veya Runtime’da düzenlenebilir olan bir integer (“n”) ile $n \times n$ ’lik bir grid oluşturmali
2. Bu grid üzerinde herhangi bir hücreye dokunulduğunda ya da tıklandığında ilgili hücreye bir X işareti yerleştirmeli
3. Oluşan X’lerden en az 3 tanesi birbirine yatay veya dikey olarak komşu hale geldiğinde ilgili X’leri yok etmelisin
4. Oyun bir kazanma ya da kaybetme mekanığıne sahip değildir
5. Oluşturulan grid daima ekrana sığmalıdır
6. Grid için canvas kullanılmamalıdır. **(BONUS)**

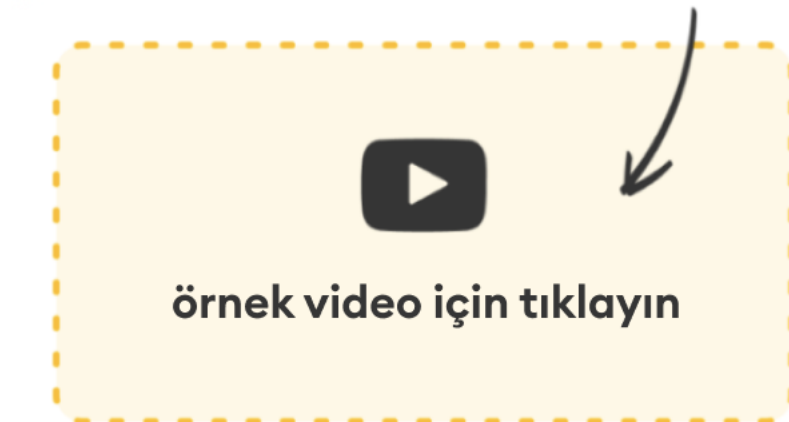


Örnek görsel üst taraftadır ve görsele bağlı kalınmak zorunda değildir; örnek $n=5$ içindir. Mevcut durumda X ‘ler kaybolmalıdır, tabi ikinci kriter verimlilik olacaktır.

PROJE 2



1. Karakter başlangıç platformundan bitiş platformuna doğru yürümektedir.
2. Yürüyeceği platformlar referans oyundakine benzer olarak soldan ya da sağdan gelerek karakterin bulunduğu platformun önünden geçmelidir.
3. Oyuncu ekrana dokunduğunda hareket halindeki platform yerine sabitlenmeli, bir önceki platformun dışına taşan kısımlar kesilerek düşüşe geçmeli ve bir sonraki kutu harekete başlamalıdır.
4. Belirli bir tolerans dahilinde kusursuz bir bir zamanlama ile platformdan bir kesilme gerçekleşmediğinde bir nota sesi oynatılmalı
5. Ard arda gerçekleşen kusursuz zamanlamalarda bu nota giderek tizleşmelidir, kusursuz seri bozulduğunda nota baştaki haline geri dönmelidir.



PROJE 2



6. Karakter daima önünde oluşan platformu ortalayacak şekilde sağa ya da sola kayarak ilerlemelidir.
7. Platform oluşamaz ve karakter aşağı düşerse oyun başarısız bir şekilde, karakter hedefe ulaşırsa başarılı bir şekilde oyun sona ermelidir.
8. Oyuncu bölümü başarılı bir şekilde tamamladığında karakter final platform üzerinde bir sevinme animasyonu gerçekleştirmeli ve bu esnada kamera karakterin etrafında dönmelidir. **(BONUS)**
9. Kamera için cinemachine kullanılmalıdır. **(BONUS)**
10. Bir bölüm tamamlandıktan sonra diğer bölüme geçerken yeni bölümün platformları bir önceki bölümün final platformunun sonuna eklenmeli ve sahne sıfırdan oluşturulmamalıdır. **(BONUS)**

KOLAYLIKLAR DILERİZ.

Challenge materyallerimizin gizliliği açısından fikri mülkiyet hakları ile ilgili dokümana [buradan](#) ulaşabilirsin.

