

## Block Game - Geliştirme Görev Listesi

Hazırlayan: Muhammet Furkan Bilici

Tarih: 8 Temmuz 2025

Bu liste, oyunu oynanabilir en basit versiyonundan (MVP) başlayarak adım adım geliştirmek için tasarlandı. Her "Part", oyunun büyük bir parçasını temsil ediyor.

### Part 1: Proje Temelleri ve Oyun Alanı

(Amaç: Boş projeden, ekranda görünen bir oyun tahtası oluşturmak.)

- TASK-01:** Unity'de yeni bir 2D projesi oluştur ve mobil platform (Android/iOS) için ayarla.
- TASK-02:** Scripts, Prefabs, Sprites, Scenes klasörlerini oluşturarak proje yapısını düzenle.
- TASK-03:** Tek bir hücreyi temsil edecek basit bir kare Sprite yarat ve bunu GridCell adıyla bir **Prefab** yap.
- TASK-04:** GridManager adında bir C# script'i ve bunu sahnede tutacak bir GameObject oluştur.
- TASK-05 [KODLAMA]:** GridManager script'i içine, oyun başladığında 8x8'lik tahtayı GridCell prefab'larını kullanarak ekrana çizen bir fonksiyon yaz.
- TASK-06 [KODLAMA]:** GridManager'a, tahtadaki hücrelerin dolu/boş durumunu takip edecek mantıksal bir yapı (örn: 2D array) ekle.

### Part 2: Blokların Oluşturulması ve Sürüklenmesi

(Amaç: Oyuncuya bloklar sunmak ve bu blokları fare/parmak ile hareket ettirebilmesini sağlamak.)

- TASK-07:** Blokların şekil verilerini (örn: T bloğu (0,0), (1,0), (-1,0), (0,-1) gibi) tutacak bir BlockData script'i veya ScriptableObject oluştur.
- TASK-08:** En az 3-4 farklı blok için BlockData asset'leri yarat.
- TASK-09:** BlockSpawner adında bir script oluştur. Bu script, rastgele 3 BlockData seçip bunları ekrandaki bekleme alanında görsel olarak oluşturmalı.
- TASK-10 [KODLAMA]:** Oyuncunun bir bloğu seçip sürüklemesini ve bırakmasını sağlayan bir BlockDragger veya benzeri bir script yaz.  
(Bu, OnMouseDown, OnMouseDown, OnMouseDown gibi fonksiyonlarla yapılabilir).

---

### Part 3: Blok Yerleştirme ve Kurallar

(Amaç: Sürüklenen bir bloğun oyun kurallarına göre tahtaya yerleşmesini sağlamak.)

- **TASK-11 [KODLAMA]:** GridManager'da, bir bloğun belirli bir koordinata yerleşip yerleşemeyeceğini kontrol eden bir fonksiyon yaz (CanPlaceBlock gibi). Bu fonksiyon, sınırları ve dolu hücreleri kontrol etmeli.
- **TASK-12 [KODLAMA]:** Blok bırakıldığında, CanPlaceBlock fonksiyonunu çağır.
  - **Eğer yerleşebiliyorsa:** Bloğu tahtaya sabitle ve GridManager'daki mantıksal grid'i güncelle.
  - **Eğer yerleşemiyorsa:** Bloğu başlangıç konumuna geri gönder.
- **TASK-13 [KODLAMA]:** Oyuncu elindeki 3 bloğu da yerleştirdiğinde BlockSpawner'ın yeni 3 blok getirmesini tetikle.

---

### Part 4: Puanlama ve Sıra Temizleme

(Amaç: Oyunun ana mekanikliği olan sıra tamamlama ve puan kazanmayı eklemek.)

- **TASK-14 [KODLAMA]:** GridManager'da, her blok yerleştikten sonra tüm satır ve sütunları kontrol edip dolu olanları bulan bir fonksiyon yaz.
- **TASK-15 [KODLAMA]:** Bulunan dolu hatları temizleyen bir fonksiyon yaz. (Görsel olarak yok et ve mantıksal grid'i "boş" olarak güncelle).
- **TASK-16:** ScoreManager adında bir script ve skoru ekranda gösterecek basit bir UI Text elemanı oluştur.
- **TASK-17 [KODLAMA]:** Her hat temizlendiğinde ScoreManager'a puan eklemesi için sinyal gönder ve UI'ı güncelle.
- **TASK-18 [KODLAMA]:** Combo mekanikliğinin ilk versiyonunu ekle: Aynı anda birden fazla hat temizlenirse bonus puan ver.

---

### Part 5: Oyun Döngüsü ve Arayüz

(Amaç: Oyunun bir sonu olmasını sağlamak ve temel menüleri eklemek.)

- **TASK-19 [KODLAMA]:** Yeni bloklar geldiğinde, oyuncunun geçerli bir hamlesi kalıp kalmadığını kontrol eden bir mantık yaz.
- **TASK-20:** Eğer geçerli hamle yoksa, "Oyun Bitti" ekranını göster. Bu ekran "Tekrar Oyna" ve "Ana Menü" butonlarını içermeli.

- **TASK-21:** Basit bir "Ana Menü" sahnesi oluřtur. Bu sahnede bir "Oyna" butonu bulunsun.
- **TASK-22:** Oyuncunun en yüksek skorunu cihaz hafızasına kaydeden ve okuyan bir sistem ekle (PlayerPrefs bunun için ideal).