

# ZOMBIE APOCALYPSE

## Top-Down Shooter Oyunu - Proje Raporu

### İçindekiler

1. Giriş
2. Sistem Gereksinimleri ve Kurulum
3. Derleme ve Oyun Kılavuzu
4. Sistem Mimarisi
5. Sınıf Yapıları ve Paketler
6. OOP Kavramlarının Uygulanması
7. Öne Çıkan Özellikler
8. Ekran Görüntüleri
9. Sonuç

### 1. Giriş

Bu proje, Java Swing kullanılarak geliştirilen kuş bakışı bir atış oyunudur (top-down shooter). Oyuncu, dalga dalga gelen çeşitli zombi türlerine karşı hayatı kalmaya çalışır. Fare ile hedef alarak farklı silahlar kullanıp zombileri öldürmek ve mümkün olduğunda yüksek puan toplamak amaçlanır.

Oyun, nesne yönelimli programlama (OOP) ilkelerine uygun olarak tasarlanmış ve çeşitli sınıf/arayüz yapıları kullanılarak modüler bir yapıda geliştirilmiştir. Proje geliştirme sürecinde, derste öğrenilen OOP kavramları (inheritance, polymorphism, encapsulation, interfaces, abstract classes) kapsamlı şekilde uygulanmıştır.

### 2. Sistem Gereksinimleri ve Kurulum

#### Sistem Gereksinimleri

- Java 8 veya üzeri (JDK 1.8+)
- Minimum 2GB RAM
- 100MB boş disk alanı

#### Kurulum

Oyun iki formatta sunulmaktadır:

1. Kaynak Kod (Derlenebilir)
  - source.zip dosyasını indirin ve bir klasöre çıkartın
  - Terminal/komut istemi kullanarak kaynak kodu derleyin
  - java Game komutu ile oyunu başlatın
2. Çalıştırılabilir Dosya (Executable)
  - ZombieApocalypse.exe dosyasını indirin
  - Direkt çalıştırarak oyunu başlatın

Not: Exe versiyonu Launch4j kullanılarak paketlenmiştir ve Windows işletim sistemlerinde çalışır. Exe dosyası Java 8 JRE gerektirir ve bilgisayarda kurulu değilse otomatik olarak indirme yönlendirilmesi yapar.

### 3. Derleme ve Oyun Kılavuzu

#### Oyunu Derleme

1. Kaynak kodun bulunduğu source.zip dosyasını bir klasöre çıkartın.
2. Bir terminal veya komut istemi açın ve çıkartığınız klasöre gidin.
3. Projeyi derlemek için:
  4. Copy
  5. javac Bil211Game2/Game/Main/Game.java
6. Derledikten sonra, oyunu çalıştmak için:
  7. Copy
  8. java Bil211Game2.Game.Main.Game

#### Oyun Kontrolleri

- W, A, S, D: Karakteri hareket ettirme (Yukarı, Sol, Aşağı, Sağ)
- W, S: Menülerde yukarı/aşağı hareket etme
- Fare: Hedef alma
- Sol Fare Tuşu: Ateş etme
- 1-5: Silah değiştirme (1-Pistol, 2-Rifle, 3-Shotgun, 4-Sniper, 5-Rocket Launcher)
- R: Şarjör doldurma
- P: Oyunu duraklatma
- H: En yakın zombiyi tespit etme
- T: FPS göstergesini açma/kapama
- ENTER: Menülerde Seçim Yapma

#### Oyun Hedefi

- Dalga dalga gelen zombileri öldürerek puan toplayın.
- Her dalga tamamlandıktan sonra yeni ve daha güçlü silahlar açılır.
- Ölen her zombi puan kazandırır, dalga ilerledikçe puan değeri artar.
- Oyun, karakter öldüğünde sona erer.

### 4. Sistem Mimarisi

Oyun mimarisi, modüler bir yapıda OOP prensiplerini uygulayan çeşitli paketler ve sınıflar içerir. Ana paketler şunlardır:

- Main: Oyunun ana döngüsü, oyun durumu kontrolü ve temel sistemler
- Entity: Oyuncular, düşmanlar ve diğer canlı varlıklar
- Objects: Silahlar, mermiler, animasyonlar ve diğer oyun nesneleri
- Tile: Oyun haritası ve çevre elementleri

Bu modüler yaklaşım, oyunun farklı bileşenlerinin birbirinden bağımsız olarak geliştirilmesine ve bakımının yapılmasına olanak sağlar.

## 5. Sınıf Yapıları ve Paketler

### Ana Paketler

Oyun kodu, mantıksal bölmelere ayrılmış beş ana paket içerir:

1. Bil211Game2.Game.Main
  - Game: Uygulamanın giriş noktası, ana JFrame oluşturur ve GamePanel'ı başlatır.
  - GamePanel: Oyunun ana döngüsünü, render işlemlerini ve oyun durumu değişikliklerini yönetir. Canvas sınıfından türetilmiş ve Runnable arayüzüne implementa etmiştir.
  - KeyHandler & MouseHandler: Kullanıcı girdisini yakalayan ve işleyen sınıflar.
  - CollisionChecker: Çarpışma tespiti işlemlerini yönetir.
  - AssetSetter: Haritaya zombileri ve nesneleri yerleştirir.
  - EventHandler & EventRect: Oyundaki özel olayları ve etkileşimleri yönetir.
  - Sound: Ses efektlerini ve müzikleri çalan sınıf.
  - UI: Kullanıcı arayüzü ve menüleri oluşturan sınıf.
  - GameSaveData: Oyun kayıt verilerini tutan sınıf.
  - UtilityTool: Yardımcı metotları içeren statik sınıf.
  - TileManager: Harita ve çevre elemanlarını yöneten sınıf.
2. Bil211Game2.Game.Entity
  - Entity: Tüm canlı varlıkların temel sınıfı (abstract).
  - Player: Oyuncunun kontrol ettiği karakteri temsil eden sınıf.
  - Monster: Tüm düşman tiplerinin temel sınıfı (abstract).
  - Zombie: Tüm zombi tiplerinin temel sınıfı (abstract).
  - NormalZombie, ReptileZombie, TankZombie, AcidZombie: Farklı zombi tipleri.
  - Acidity, Jumpable: Farklı zombi yeteneklerini tanımlayan arayüzler.
3. Bil211Game2.Game.Objects.Ammos
  - Ammo: Tüm mermi tiplerinin temel sınıfı (abstract).
  - PistolAmmo, RifleAmmo, ShotgunAmmo, SniperAmmo, RocketAmmo, Acid: Farklı mermi tipleri.
4. Bil211Game2.Game.Objects.Weapons
  - Weapon: Tüm silahların temel sınıfı (abstract).
  - Pistol, Rifle, Shotgun, Sniper, RocketLauncher: Farklı silah tipleri.
5. Bil211Game2.Game.Objects.Animations
  - Animation: Tüm animasyonların temel sınıfı.
  - AnimationManager: Animasyonları yöneten sınıf.
  - ExplosionAnimation, AcidSplashAnimation, BloodSplashAnimation: Farklı animasyon tipleri.

## 6. OOP Kavramlarının Uygulanması

### 1. Inheritance (Kalıtım)

Oyun, kalıtım yapısını etkin şekilde kullanmaktadır. Örneğin:

- Entity sınıfından Player ve Monster sınıfları türetilmiştir
- Monster sınıfından Zombie sınıfı türetilmiştir

- Zombie sınıfından NormalZombie, ReptileZombie, TankZombie ve AcidZombie sınıfları türetilmiştir
- Ammo sınıfından farklı mermi tipleri türetilmiştir
- Weapon sınıfından farklı silah tipleri türetilmiştir

Bu kalıtım yapısı, kodun yeniden kullanılabilirliğini artırır ve her alt sınıfın özelleştirilmiş davranışlara sahip olmasına olanak tanır.

## 2. Polymorphism (Çok Biçimlilik)

Polymorphism, farklı sınıfların ortak bir arayüz üzerinden kullanılabilmesini sağlar. Oyunda:

- Entity sınıfının update ve draw metodları, farklı entity türleri için özelleştirilmiştir
- Monster sınıfının kill metodu her zombi türünde farklı davranış sergiler
- Weapon sınıfının fire metodu her silah türünde farklı işlevler gerçekleştirir

GamePanel sınıfı tüm entityleri günceller ve çizer, her entity'nin kendi türüne göre özelleştirilmiş update ve draw metodlarını çağırır.

## 3. Encapsulation (Kapsülleme)

Kapsülleme, sınıfların iç detaylarını gizleyerek sadece gerekli metodları ve özellikleri dışarıya açar. Örneğin:

- Entity sınıfındaki birçok özellik private veya protected olarak tanımlanmıştır
- Getter/setter metodları kullanılarak kontrollü erişim sağlanmıştır
- Her sınıf, kendi iç işlevlerini diğer sınıflardan gizlemektedir

## 4. Abstraction (Soyutlama)

Soyutlama, karmaşık sistemleri basitleştirmek temel özelliklere odaklanmayı sağlar. Oyunda:

- Entity, Monster, Weapon ve Ammo gibi abstract sınıflar kullanılmıştır
- Bu sınıflar, alt sınıfların uygulaması gereken soyut metodlar tanımlar
- Örneğin, tüm Monster sınıfları kill() metodunu uygulamak zorundadır, ancak her biri bunu kendi türüne özgü bir şekilde yapar

AcidZombie sınıfının kill() metodunda, oyuncuya puan ekleme, asit sıçratma, kan efekti oluşturma ve loot düşürme işlemleri yapılır.

## 5. Interfaces (Arayüzler)

Oyunda, sınıfların belirli davranışları uygulamasını zorunlu kılmak için arayüzler kullanılmıştır:

- Acidity: Asit sıçratma yeteneği olan varlıklar için (AcidZombie sınıfı)
- Jumpable: Ziplama yeteneği olan varlıklar için (ReptileZombie sınıfı)

ReptileZombie sınıfı, Jumpable arayüzü uygulayarak ziplama yeteneğine sahip olur.

Bu arayüzler, ilgili yetenekleri gerektiren sınıfların belirli metodları uygulamasını sağlayarak, kodun modülerliğini ve esnekliğini artırır.

## 7. Öne Çıkan Özellikler

### 1. Olay Yönetim Sistemi (Event System)

Oyunda, çeşitli olayları ve etkileşimleri yönetmek için kapsamlı bir Event sistemi bulunmaktadır:

- EventHandler: Oyun içi olayları yöneten ana sınıf
- EventRect: Olay alanlarını tanımlayan dikdörtgen sınıf
- Çeşitli Olay Tipleri: Teleport, hasar, item bulma vb.

EventHandler sınıfı, haritada tanımlanan belirli alanlara (EventRect) oyuncu girdiğinde çeşitli olayları tetikler. Bu olaylar arasında:

- Hasar Çukurları: Oyuncu üzerine bastığında can kaybına neden olan alanlar
- Teleport Noktaları: Oyuncuyu haritanın bir bölgesinden başka bir bölgesine işinlama

Bu sistem, oyunun çevre ile etkileşimini zenginleştirir.

### 2. Gelişmiş Ses Sistemi

Oyunda kapsamlı bir ses sistemi bulunmaktadır. Sound sınıfı, ses efektlerini ve müzikleri yönetmek için ayrı bir thread kullanır.

Sound sınıfı, sesleri yönetmek için bir komut kuyruk yapısı kullanır. Bu yapı, ses komutlarını (müzik yükleme, efekt çalma, ses seviyesi ayarlama) bir kuyrukta biriktirir ve arka planda çalışan bir thread tarafından işler.

Ses sistemi şunları içerir:

- Arka plan müziği (menü ve oyun için farklı müzikler)
- Her silah için özel ateş sesleri
- Zombi ölüm sesleri
- Oyuncu hasar sesleri
- Patlama ve asit efektləri
- Dalga tamamlama sesi

### 3. Ekran Sallama Efekti

Oyuna gerçekçilik katmak için özel bir ekran sallama efekti eklenmiştir. Bu efekt, oyuncu hasar aldığında veya patlama olduğunda devreye girer.

startScreenShake metodu, sallamanın yoğunluğunu ve süresini belirler. updateScreenShake metodu ise her kare güncellemede ekran sallamasını hesaplar ve zamanla azalan bir yoğunlukta rastgele ofsetler oluşturur.

Bu efekt, oyuna daha fazla etki ve gerçeklik katar, özellikle patlama veya hasar alma anlarında oyuncuya görsel geri bildirim sağlar.

### 4. Zombi Tespit Sistemi

Oyuncu, H tuşuna basarak en yakın zombiyi tespit edebilir. Bu özellik, zombilerin görünüm haritada görünmediği durumlarda oyuncuya yardımcı olur.

findClosestZombie metodu, tüm zombiler arasında oyuncuya en yakın olanı bulur. Bu zombi aktif olduğunda, ekranда zombie doğru bir ok gösterilir ve bir radar efekti oluşturulur.

drawDirectionArrow metodu, en yakın zombieye doğru bir ok çizer. Okun yanı sıra, oyuncunun etrafında dalgalanan bir halka efekti de eklenir, bu da radar benzeri bir görünüm sağlar.

## 5. Kapsamlı Animasyon Sistemi

Oyun, çeşitli görsel efektler için kapsamlı bir animasyon sistemi kullanır:

- ExplosionAnimation: Roketatar patlamalarını gösterir
- AcidSplashAnimation: Asit zombileri öldürülüğünde veya asit mermisi çarptığında oluşur
- BloodSplashAnimation: Zombiler öldürülüğünde kan efekti oluşturur

AnimationManager sınıfı, bu animasyonları oluşturur ve yönetir. Her animasyon, belirli bir süre boyunca oynatılır ve tamamlandığında otomatik olarak kaldırılır.

## 6. Oyun Kaydetme ve Yükleme Sistemi

Oyun durumu, FileIO kullanılarak kaydedilebilir ve daha sonra yüklenebilir. Kayıt sistemi şunları içerir:

- Oyuncu durumu (pozisyon, can, skor)
- Zombi durumu (tür, pozisyon, can)
- Mermi durumu (tür, pozisyon, hız)
- Silah durumu (mermi sayısı, şarjör)
- Oyun fazı ve zombi şansları

GameSaveData sınıfı, kaydedilecek tüm verileri tutar. saveGame ve loadGame metotları, bu verileri binary formatta kaydeder ve yükler.

Bu sistem, oyuncuların oyunu kaydedip daha sonra kaldıkları yerden devam etmelerini sağlar.

## 7. Exe Paketleme Sistemi

Oyun, Launch4j kullanılarak Windows platformu için çalıştırılabilir (.exe) bir dosya haline getirilmiştir. Bu, Java bilgisi olmayan son kullanıcıların da oyunu kolayca çalıştırılmasını sağlar.

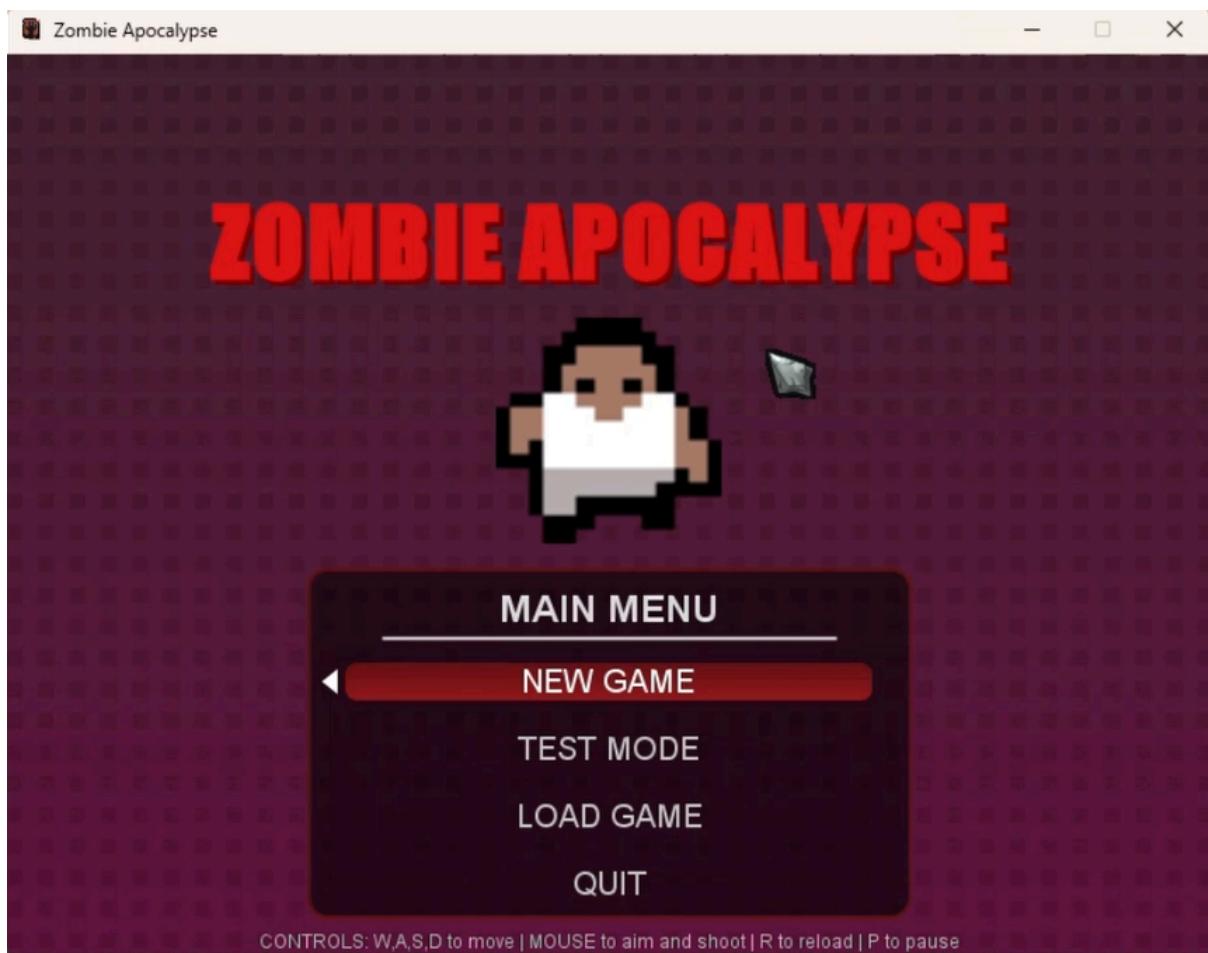
Exe paketi aşağıdaki özelliklere sahiptir:

- Java Runtime Environment (JRE) kontrolü
- Eksik JRE durumunda otomatik indirme yönlendirmesi
- Özel oyun simgesi ve splash screen
- Windows başlangıç menüsü entegrasyonu

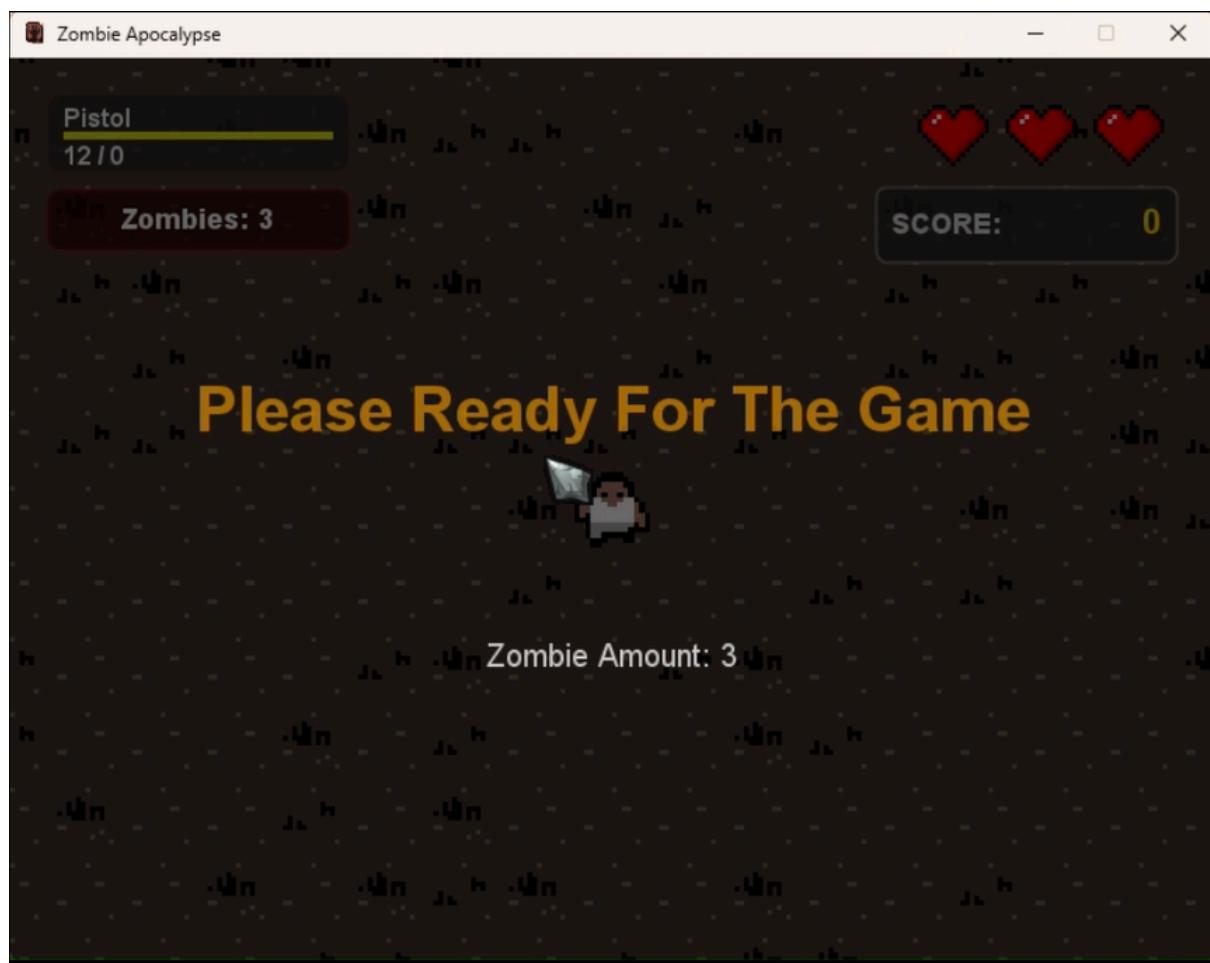
Bu sayede oyun, standart bir Windows uygulaması gibi kolayca kurulup çalıştırılabilir.

## 8. Ekran Görüntüleri

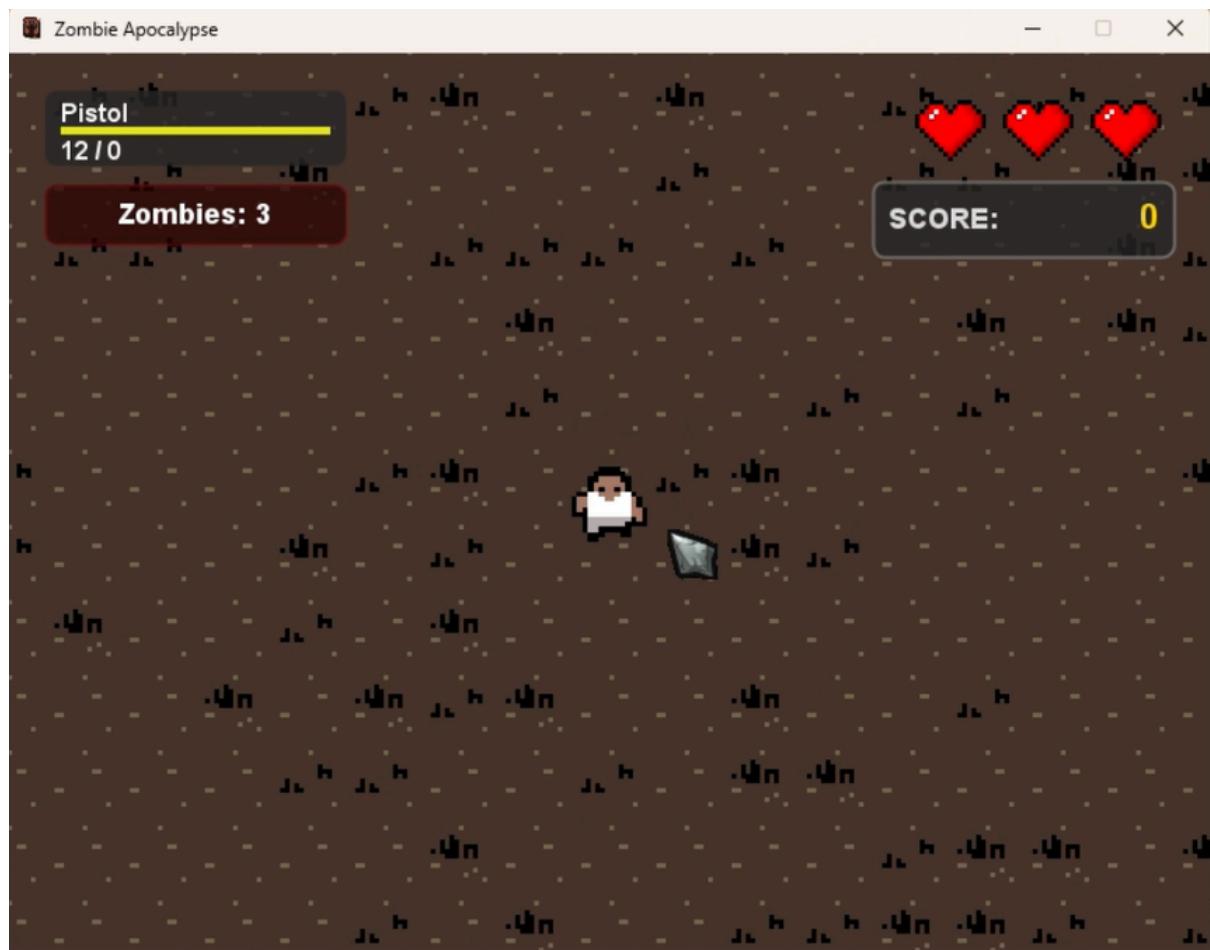
Ana Menü Ekranı



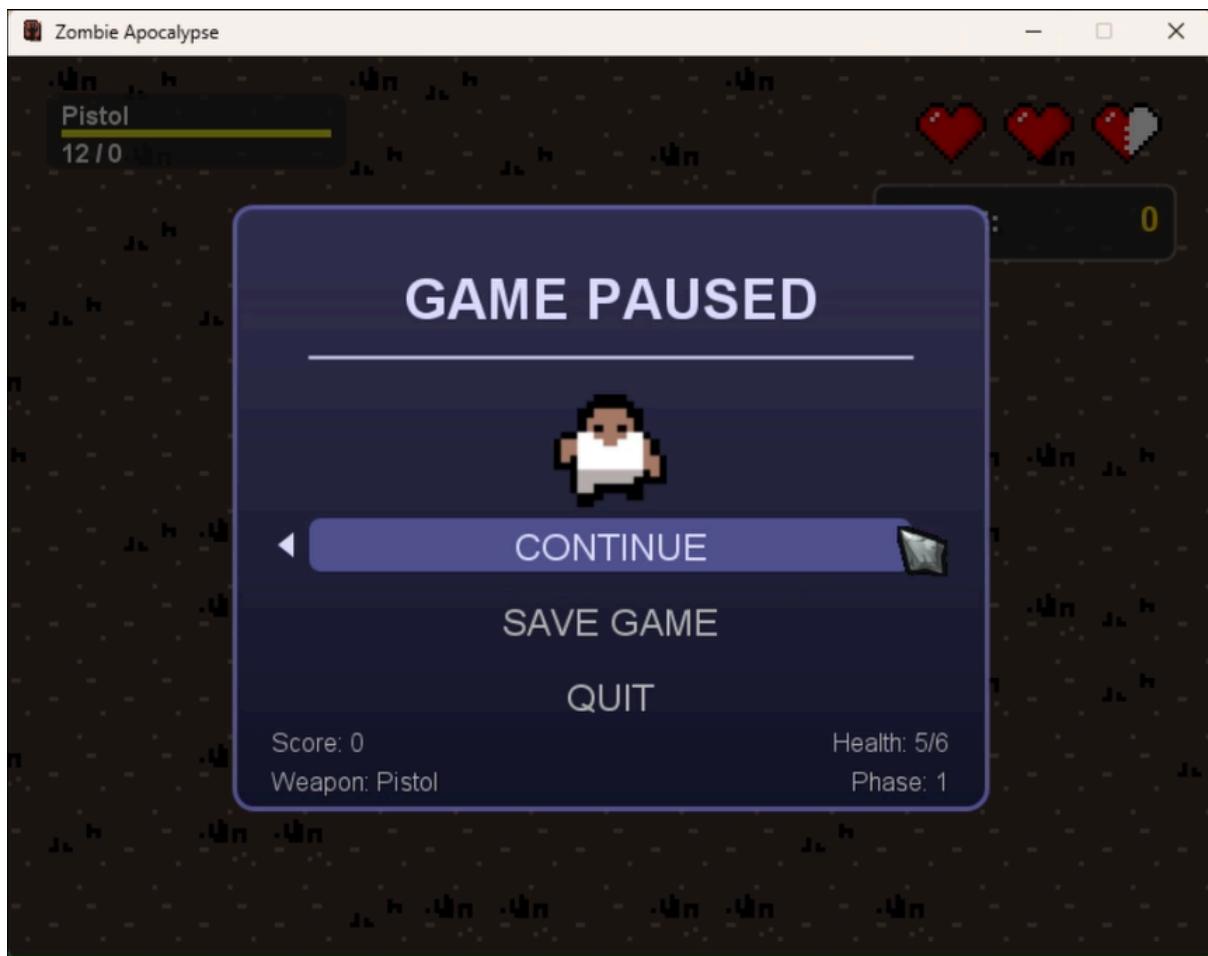
## Oyun Başlangıç Ekranı



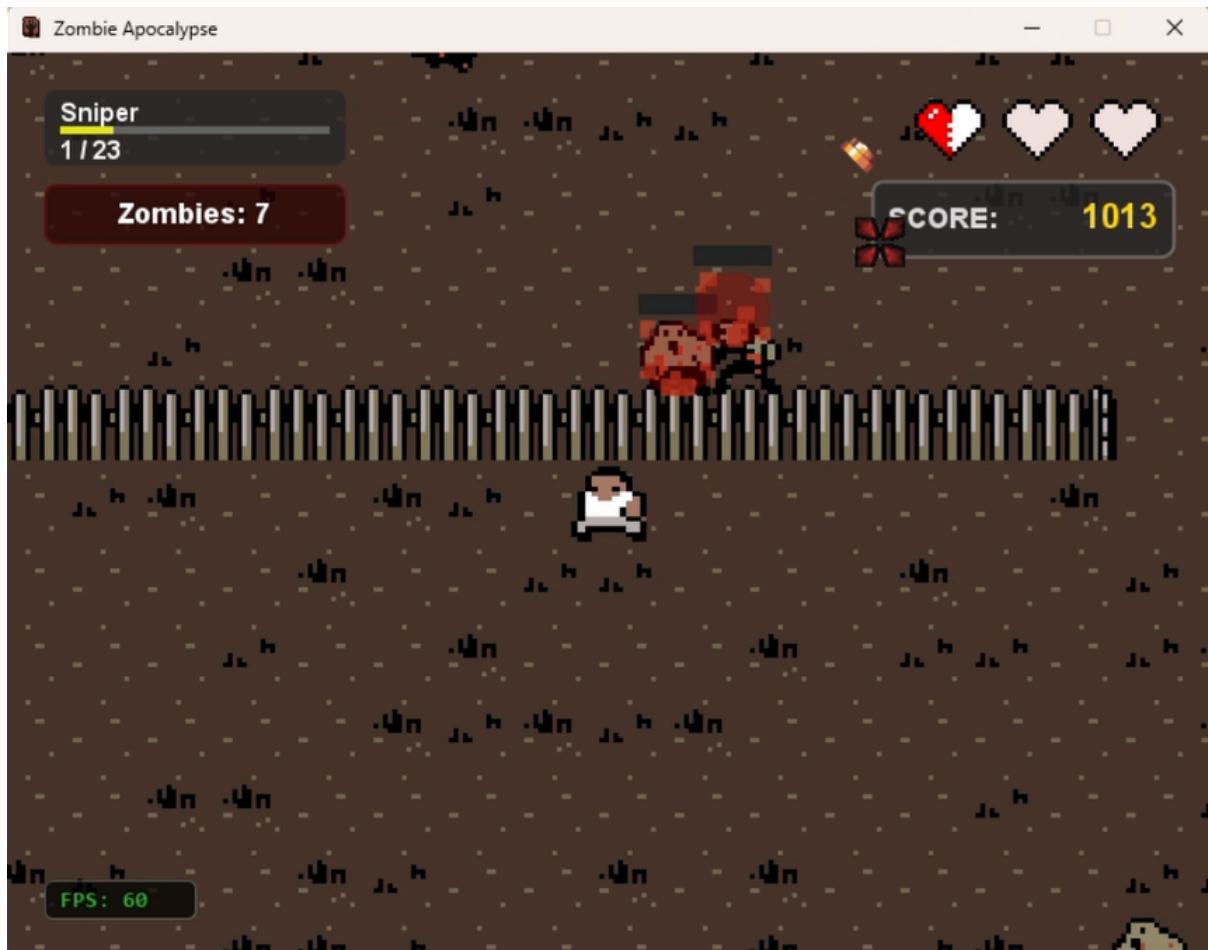
## Oyun İçi Ekran



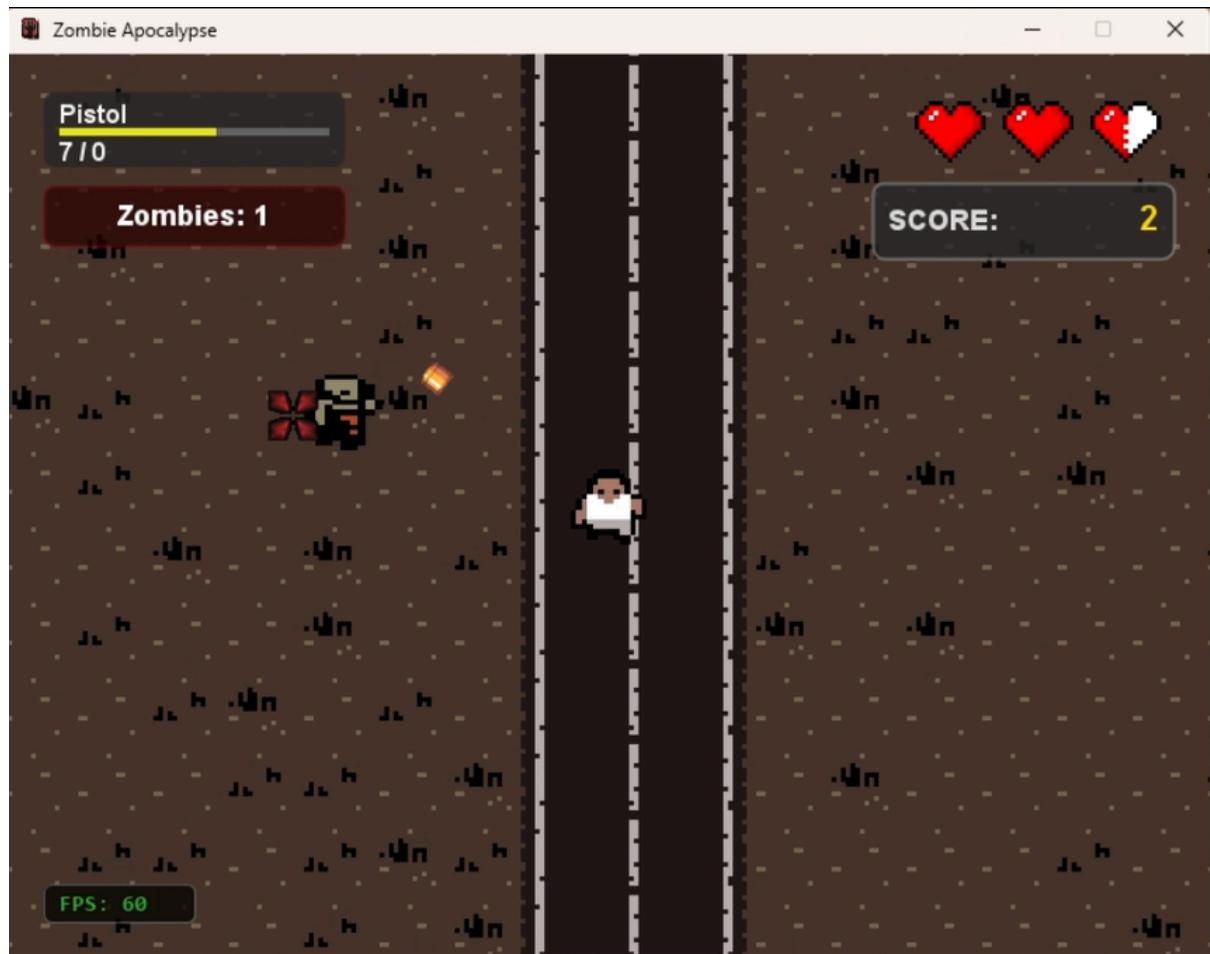
## Duraklatma Menüsü



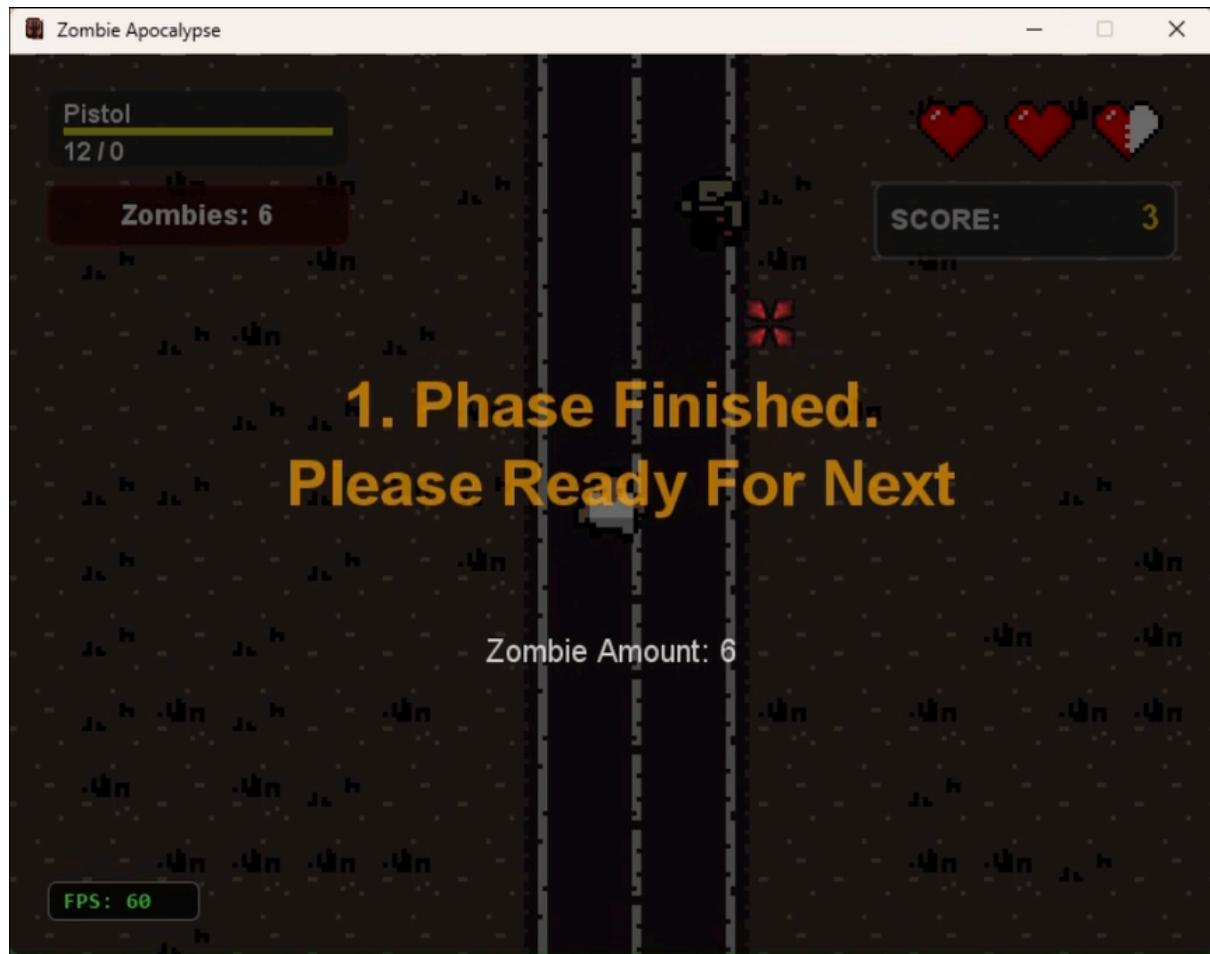
## Savaş ve Kan Efektleri



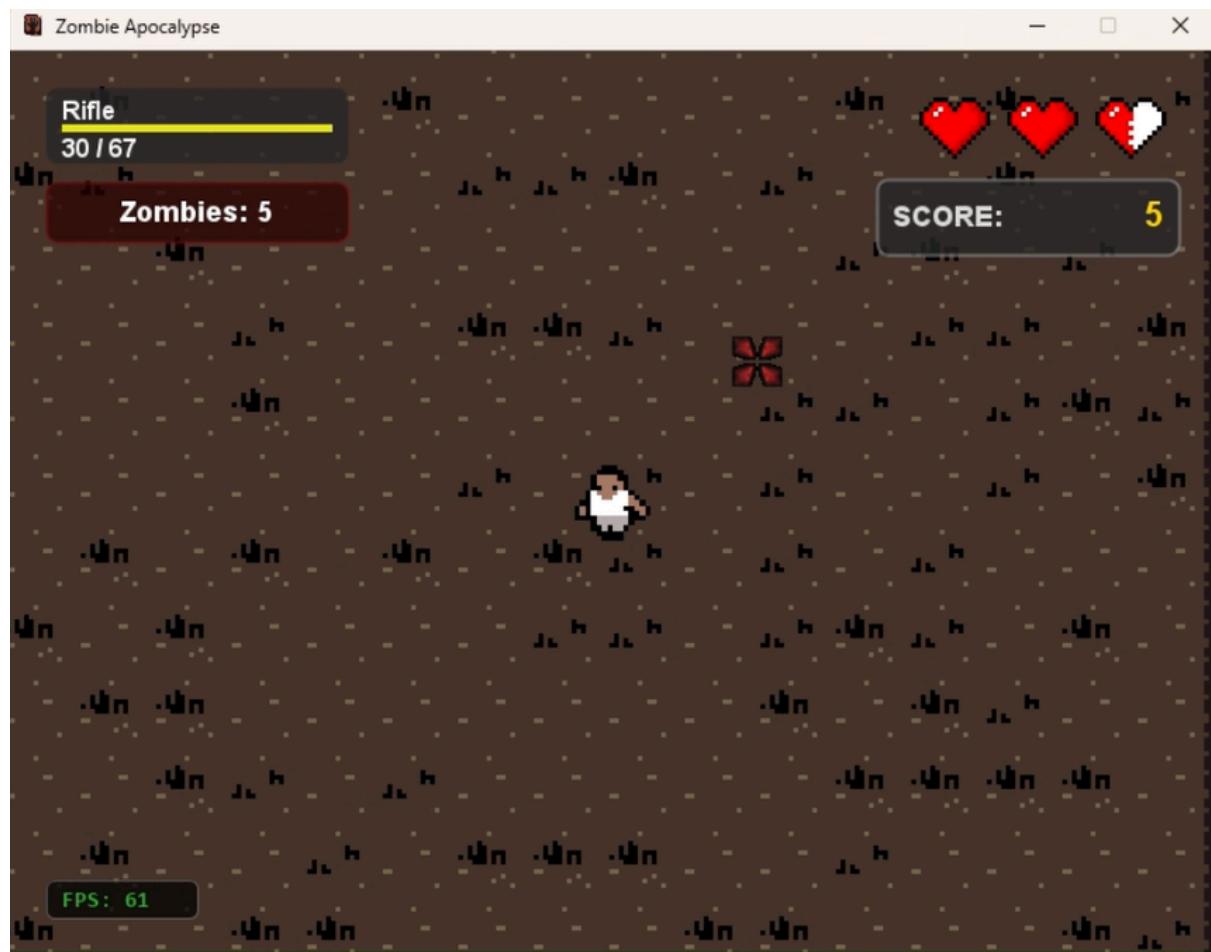
## Harita Üzerinde İlerleme



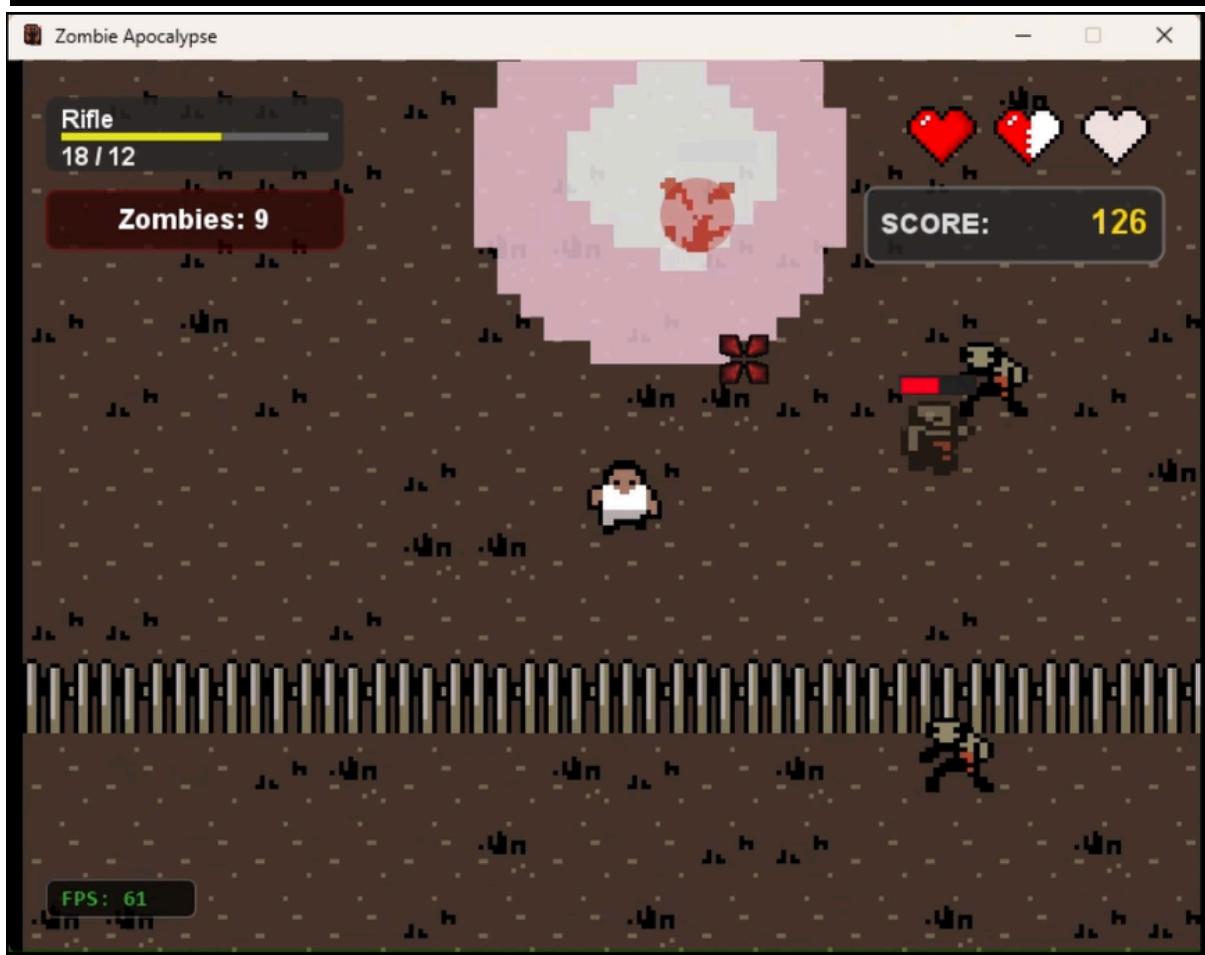
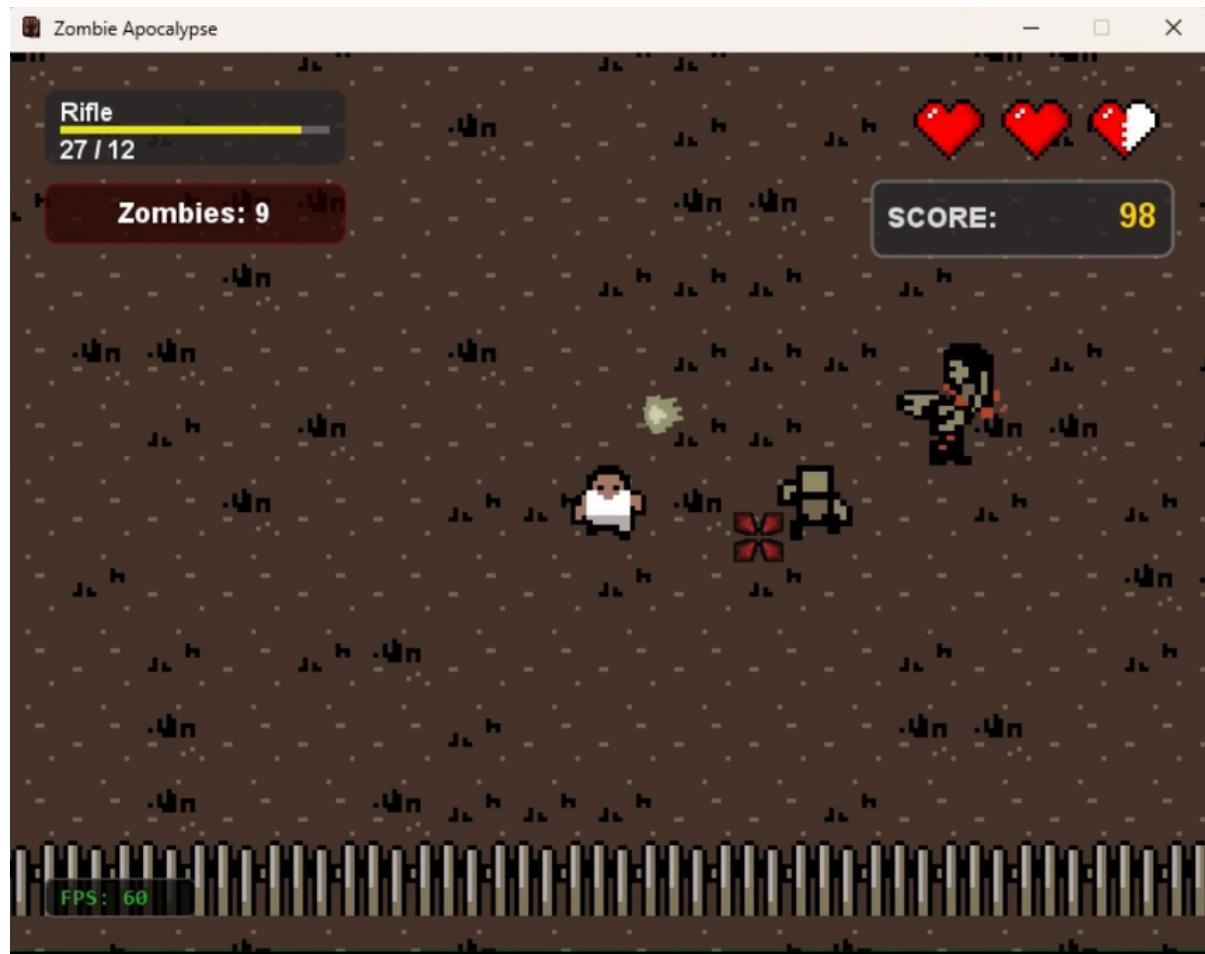
## Dalga Tamamlama Ekranı



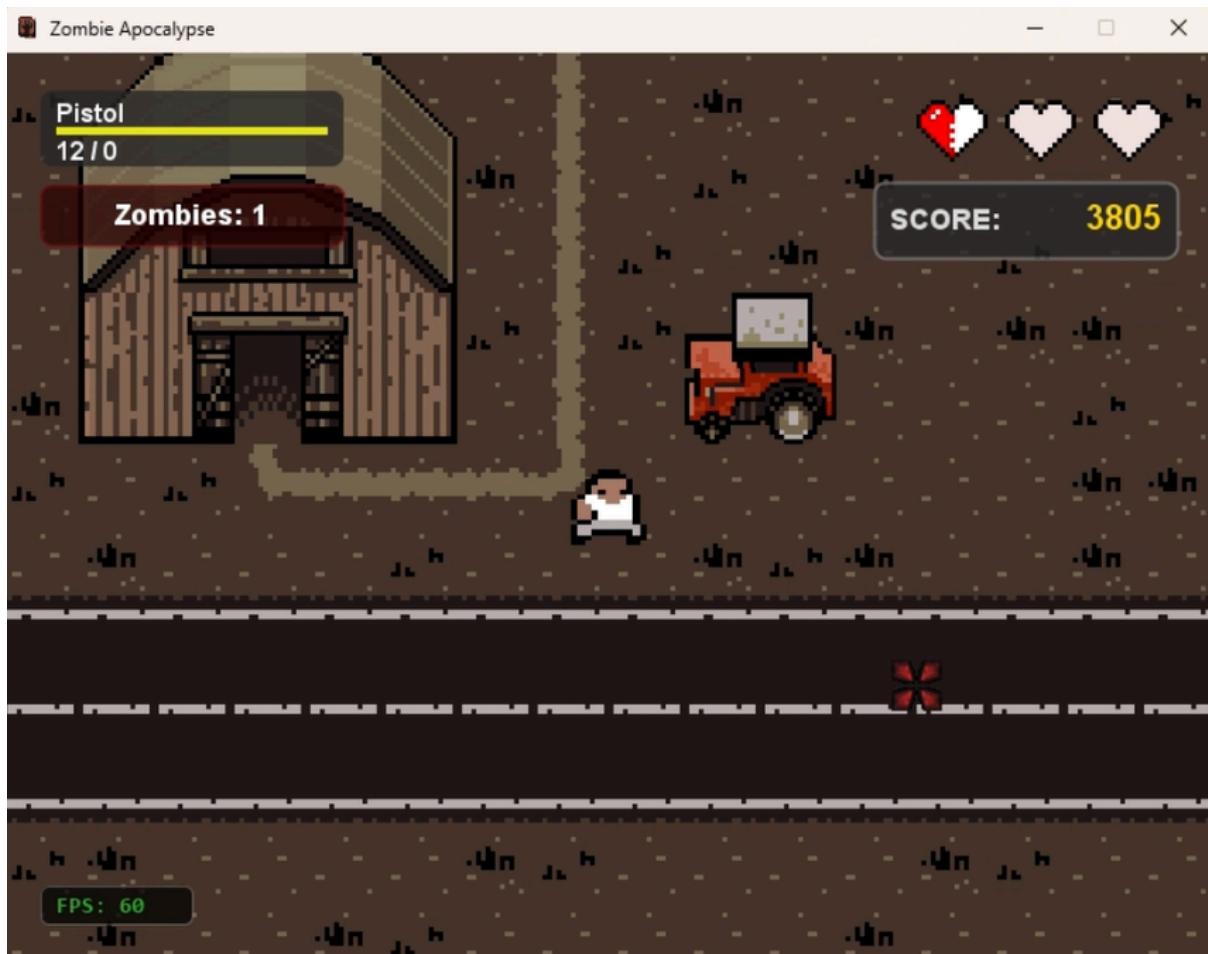
## Rifle Silahı Kullanımı



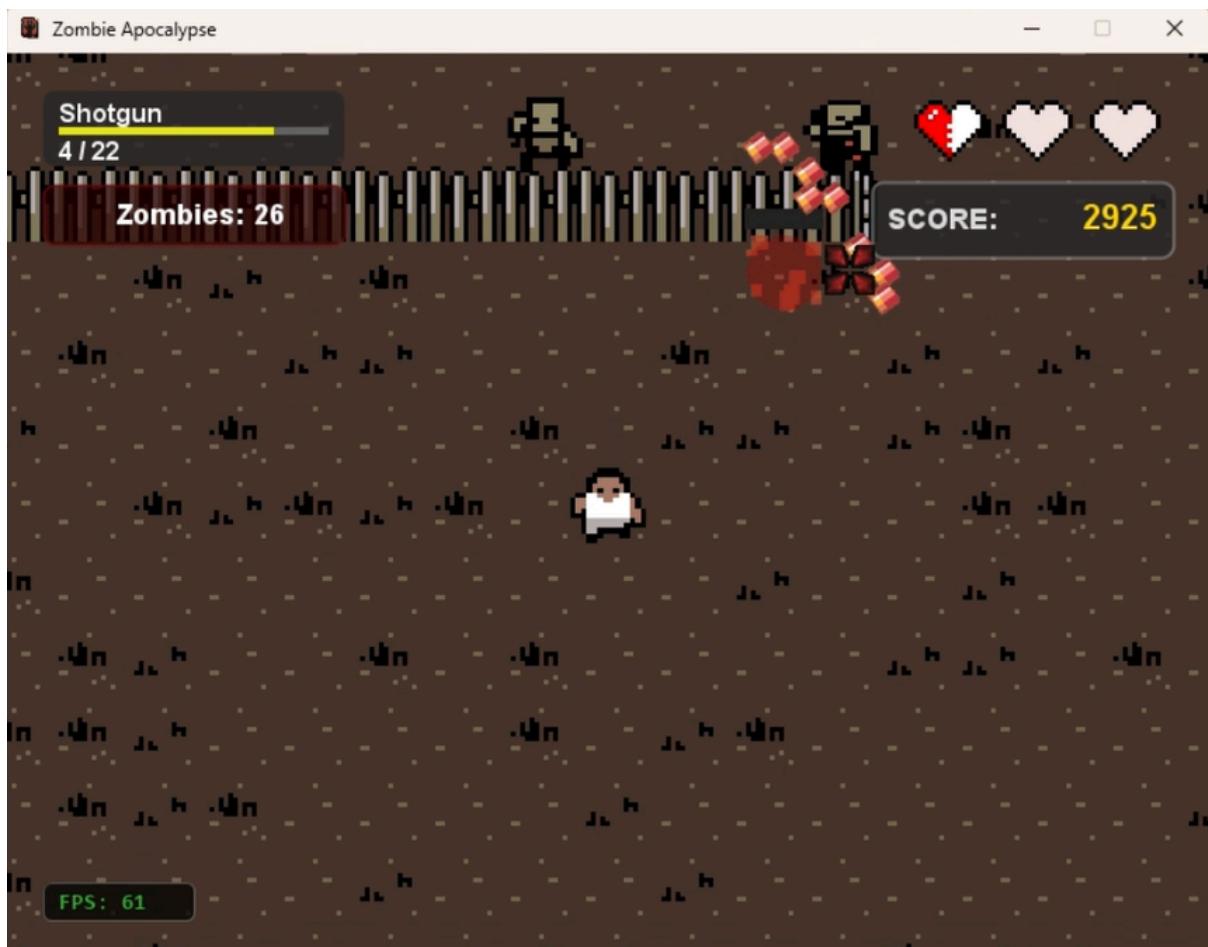
## Asit Zombi Efekti



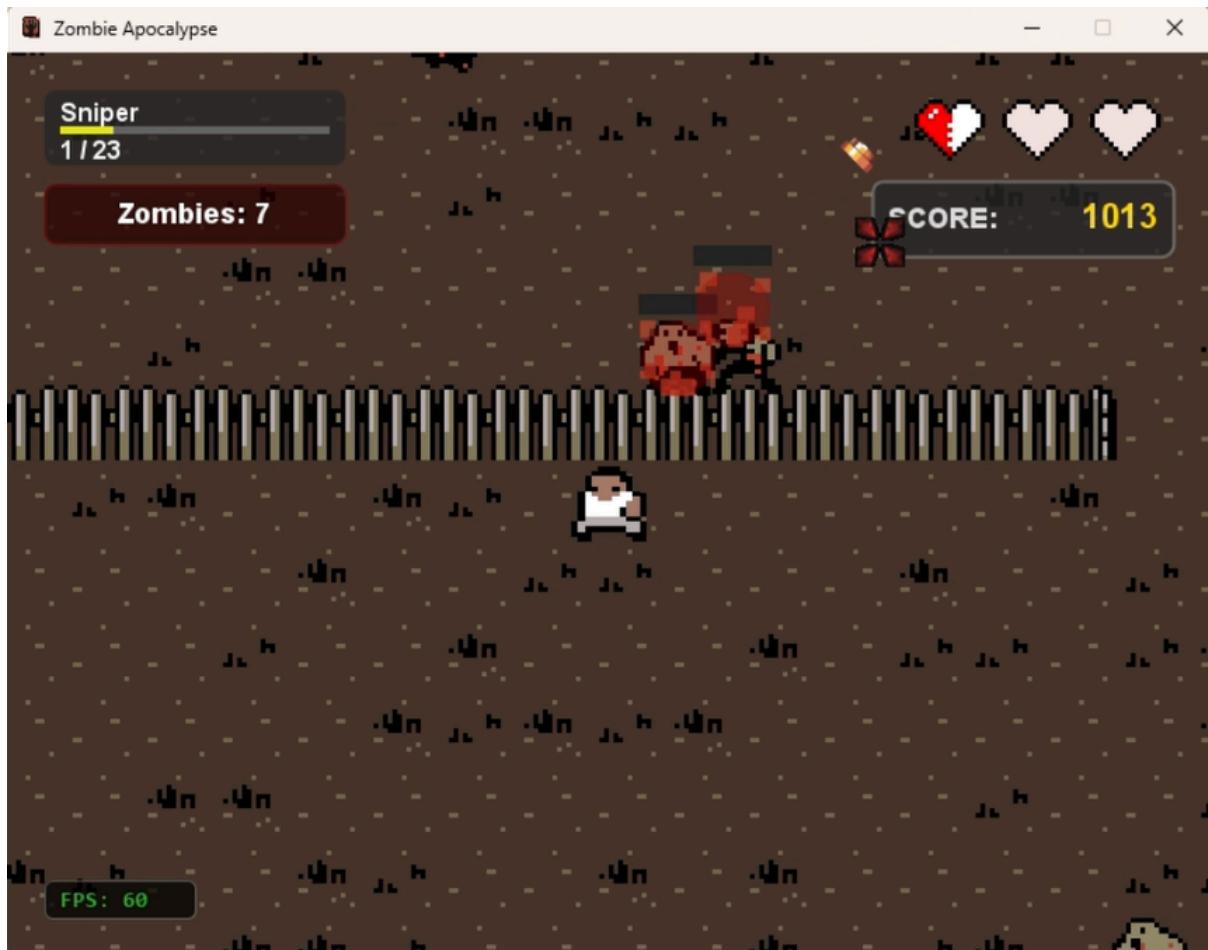
## Çiftlik Bölgesi



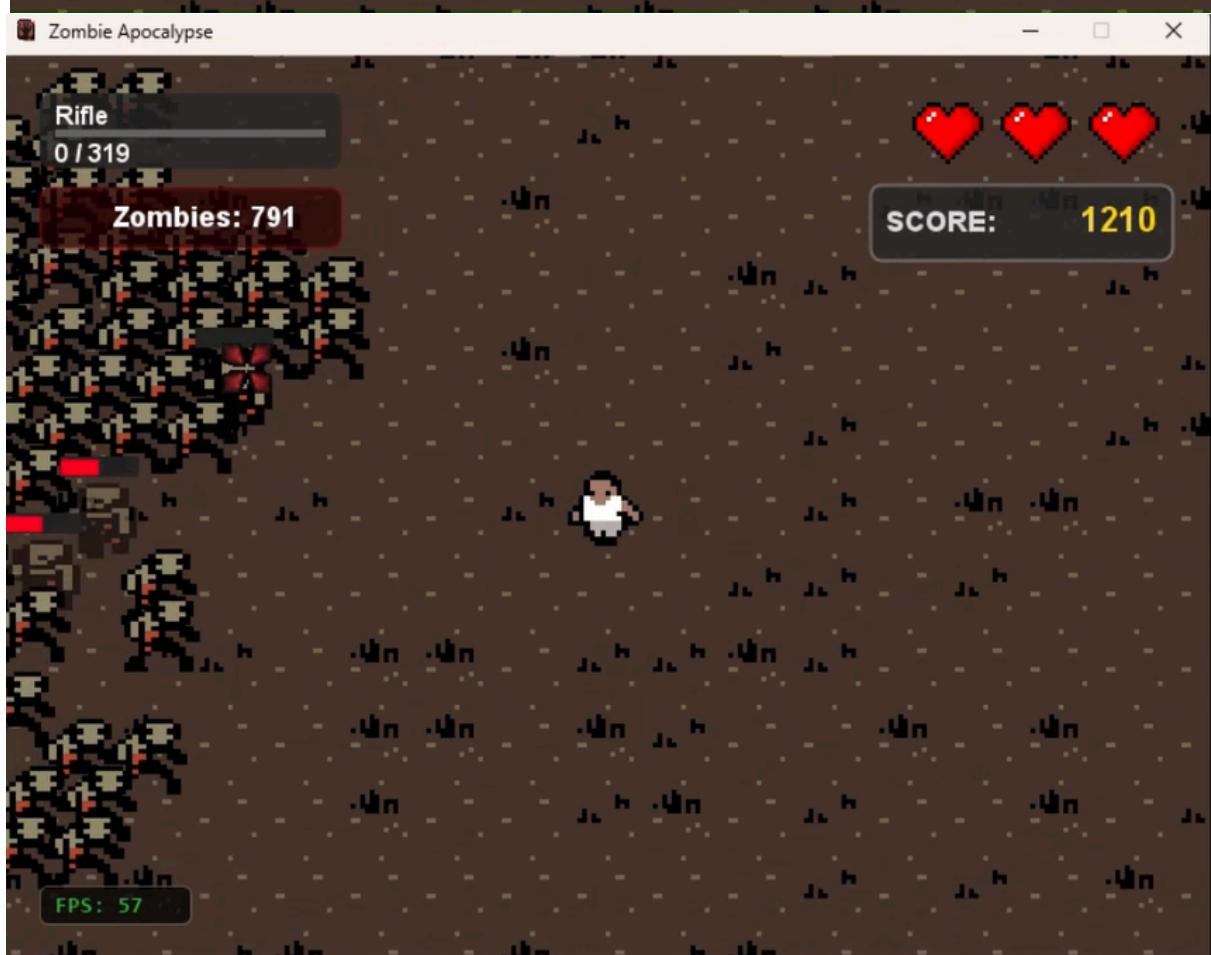
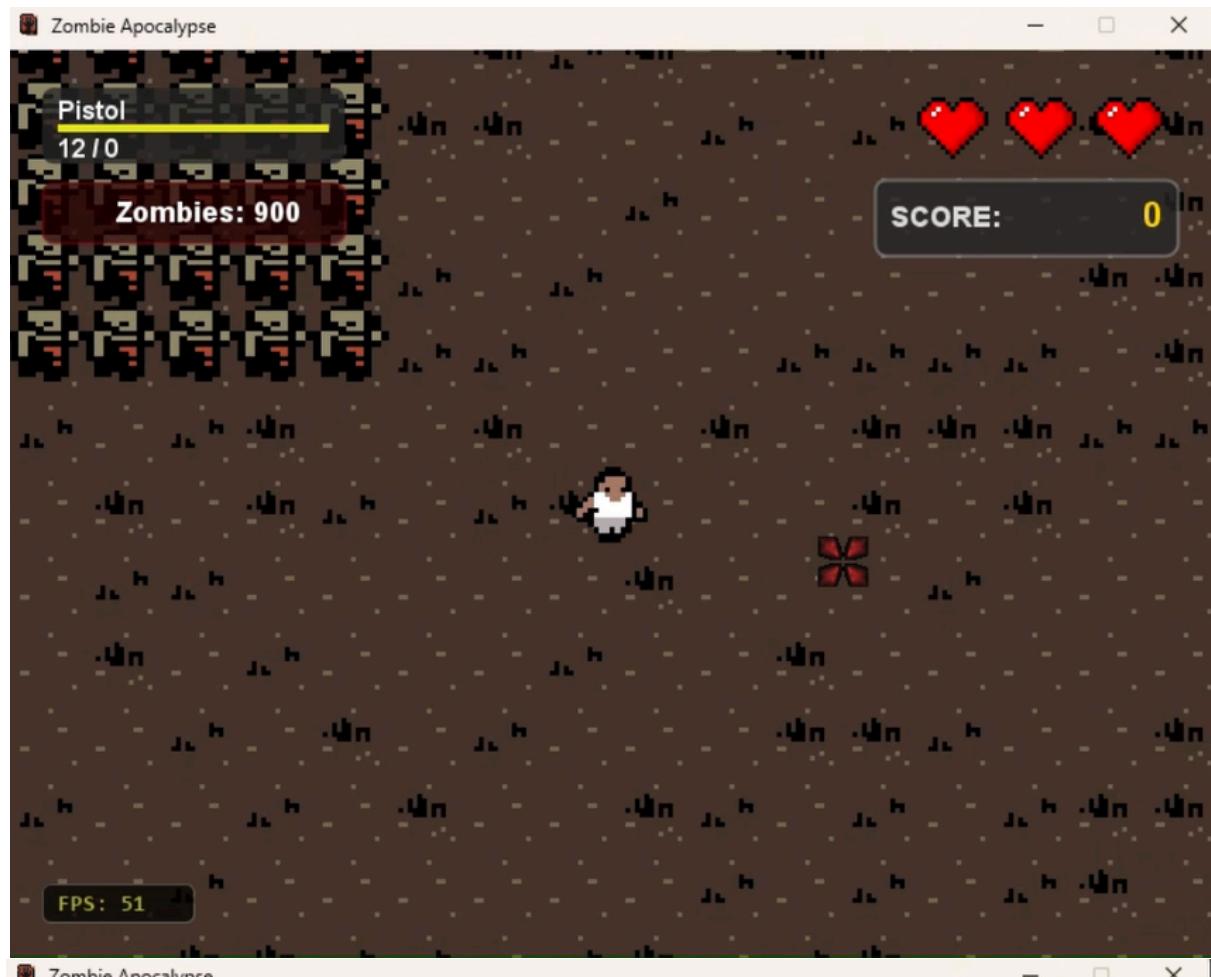
## Shotgun Saçma Efekti



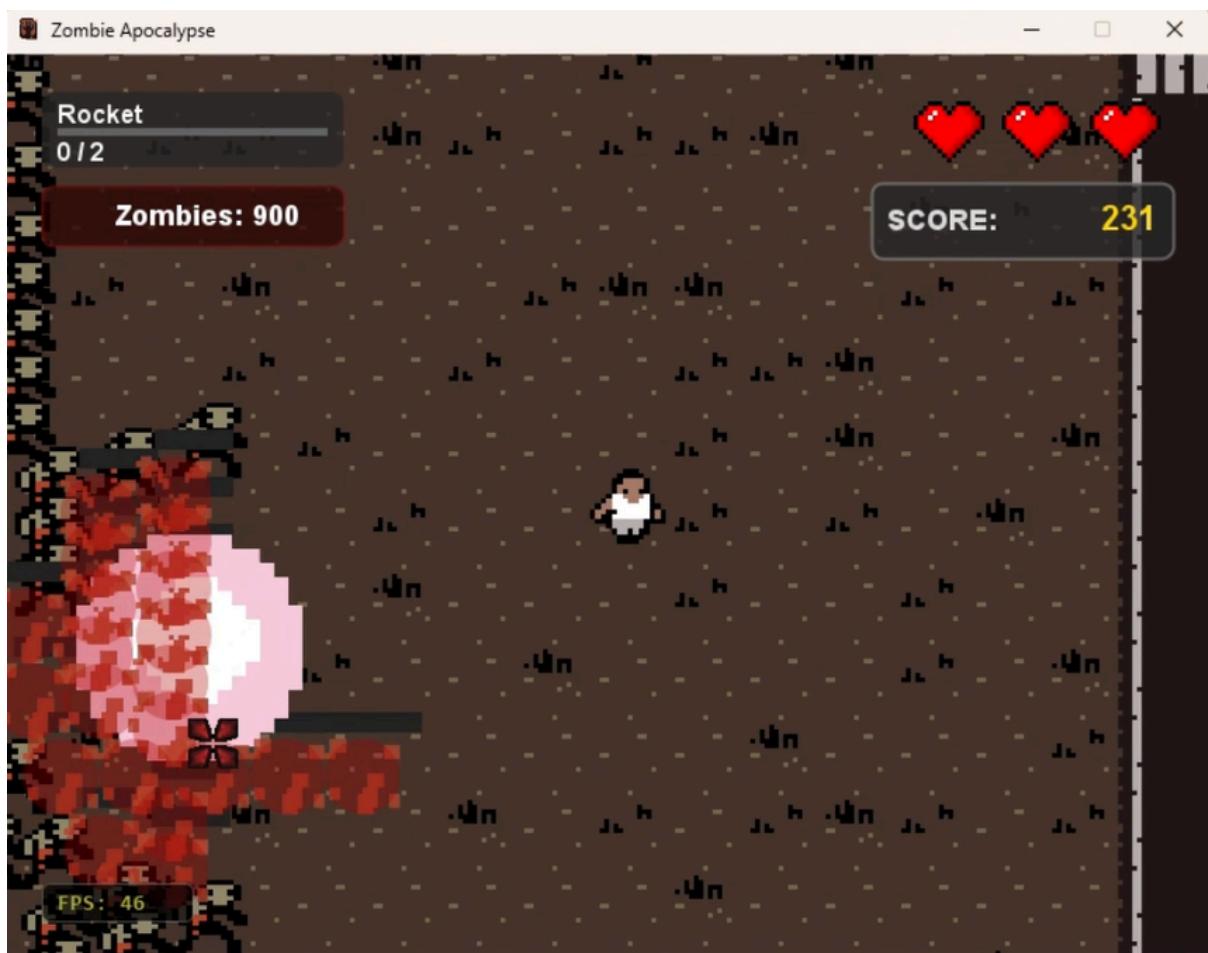
## Sniper Mermisi Efekti



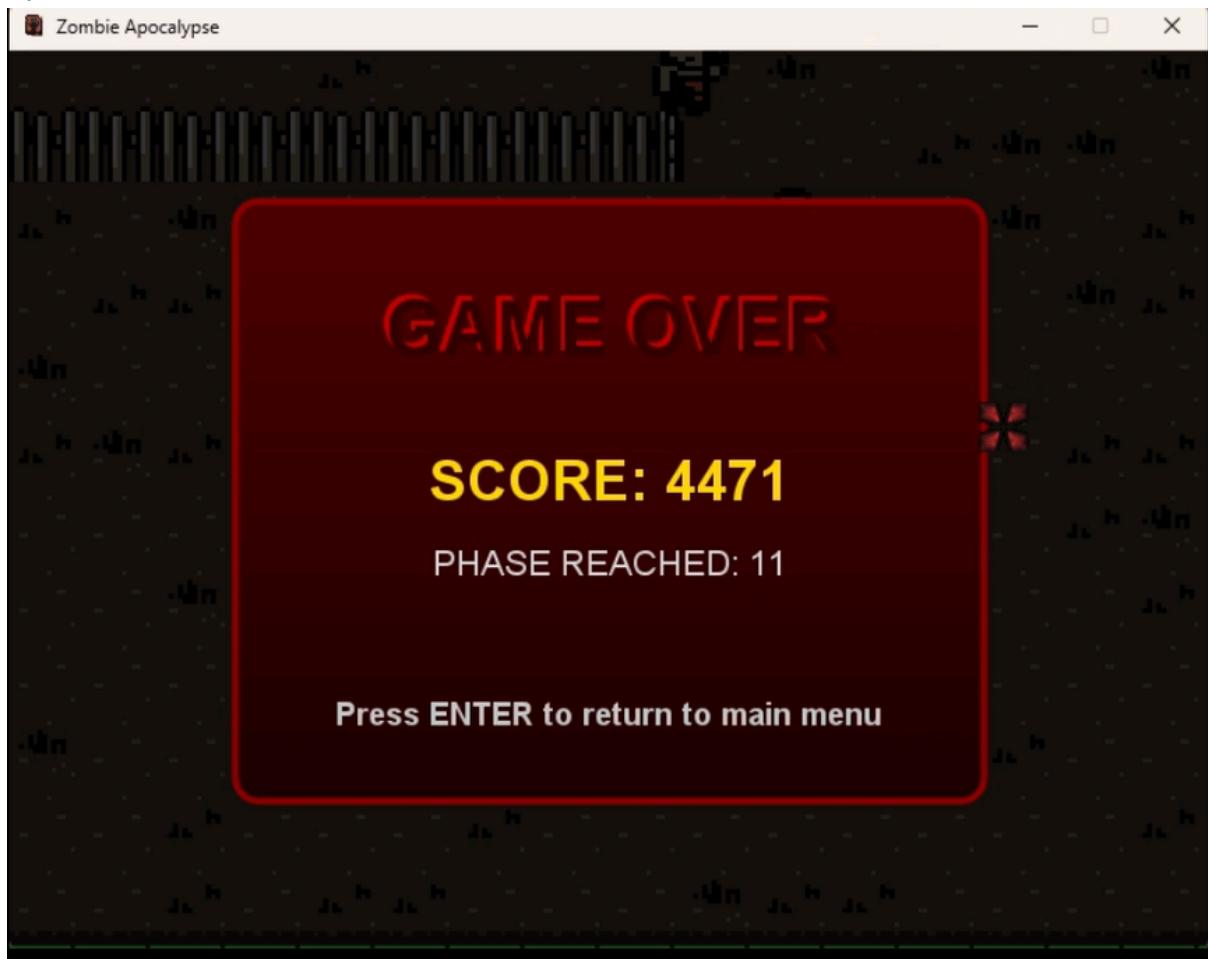
## Test Modu



## Roketatar Patlaması!



## Oyun Sonu



## 9. Sonuç

Bu proje, Java Swing kullanılarak geliştirilen kapsamlı bir top-down shooter oyunudur. OOP ilkelerini etkin bir şekilde uygulayarak modüler ve genişletilebilir bir kod yapısı oluşturulmuştur. Kalıtım, polimorfizm, kapsülleme, soyutlama ve arayüzler gibi OOP kavramları, oyunun farklı bileşenlerini yapılandırmak için kullanılmıştır.

Proje, temel oyun gereksinimlerinin ötesinde, EventHandler sistemi, ekran sallama efekti, kapsamlı ses sistemi, zombi tespit sistemi ve gelişmiş animasyon sistemi gibi ek özelliklerle zenginleştirilmiştir. Bu özellikler, oyun deneyimini daha sürükleyici ve eğlenceli hale getirmektedir.

Oyun Java 8 veya üzeri sürüm gerektirir ve hem kaynak kod olarak derlenebilir hem de Windows platformları için hazır exe dosyası olarak dağıtılabılır. Launch4j ile paketleme sayesinde son kullanıcılar için kurulum ve kullanım kolaylığı sağlanmıştır.