

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

**PROJE KONUSU:** **2023-2024 BAHAR DÖNEMİ BLM232/YZM228 PROGRAMLAMA LAB II**

**ÖDEVİ- LORDS OF THE POLYWARPHISM**

**ÖĞRENCİ ADI:Emir Dursun**

**ÖĞRENCİ NUMARASI:220502001**

**GITHUB:** **github.com/emirdrs**

**ÖĞRENCİ ADI:Serhat Arslaner**

**ÖĞRENCİ NUMARASI:220502043**

**GITHUB:** **github.com/serhatarslaner**

**DERS SORUMLUSU:**

**PROF. DR./DR. ÖĞR. ÜYESİ Prof. Dr. Nevcihan DURU**

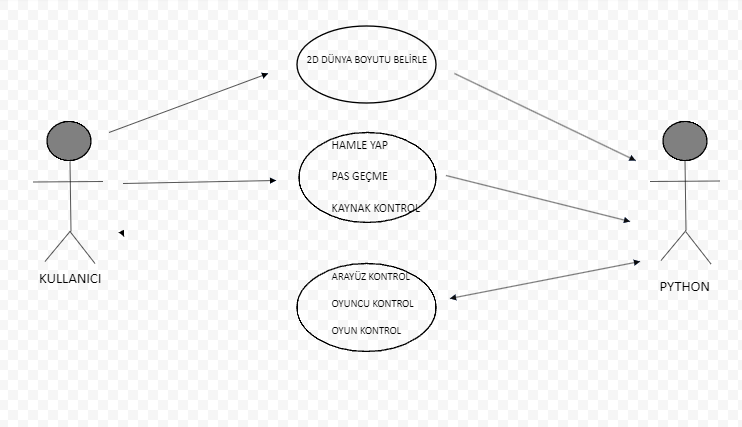
**TARİH:24.03.2024**

1. **GİRİŞ**
   1. **Projenin amacı**

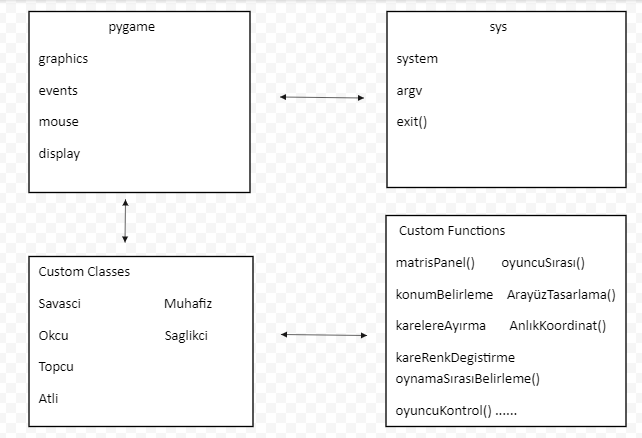
* Projemizdeki amacımız, Python dili üzerinde kullanıcı tarafından belirlenen 2 boyutlu bir matris alanı büyüklüğünde ki kareli dünyalarda çok oyunculu bir savaş-strateji oyunu geliştirmek.

1. **GEREKSİNİM ANALİZİ**
   1. **Arayüz gereksinimleri**

* Kullanıcı arayüzümüzde ilk açılışta oyuncuların bulunduğu kareli dünyada her köşeye bir oyuncu yerleştirilmiş şekilde açılacak.
* Oyun dünyasının sağ tarafındaki boş alanda Oyuncu Sırası : M2 gibi oyuncu sırasının kimde olduğunu belirten bir yazı olacak,bu yazı oyuncunun rengiyle yazılacak.
* Oyun dünyasının sağ tarafında istenilen savaşçı türlerini seçebildiğimiz,yanlarında gereken kaynak miktarı belirtilen 5 adet buton olacak.
* Aynı şekilde oyun dünyasının sağ tarafında oyuncuların pas geçebilmesi için PAS isminde bir buton olacak.
  1. **Fonksiyonel gereksinimler**
* Bir oyuncu 3 kez pas geçtiği zaman oyundan elenecek.
* Bir oyuncu sıra ondayken en fazla 2 hamle yapabilecek,sonrasında sıra diğer oyuncuya geçecek.
* Sağ tarafta bulunan ekrandan istenen savaşçı seçildikten sonra istenen kareye sağ tıklandığında o kareye o savaşçı eklenecek.
* Oyuncuların savaşçıları ekleyebileceği bölgelerde sınır,yani sadece kendilerinin veya askerlerinin 1 kare etrafına koyabilmesine izin verilecek,koyabileceği alanlar da kenarları oyuncu rengiyle boyanmış olan karelerle belirtilecek.
  1. **Use-Case diyagramı**



1. **TASARIM**
   1. **Mimari tasarım**

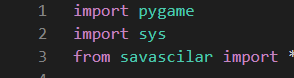


* 1. **Kullanılacak teknolojiler**
* Python kodlama dilinde yazılacak.
* Python üzerinde oyun yapımını kolaylaştıran pygame kütüphanesi,kapatma işleminden emin olmak için sys kütüphanesi.
* Visual Studio Code
* Python IDE
* Pip Modül

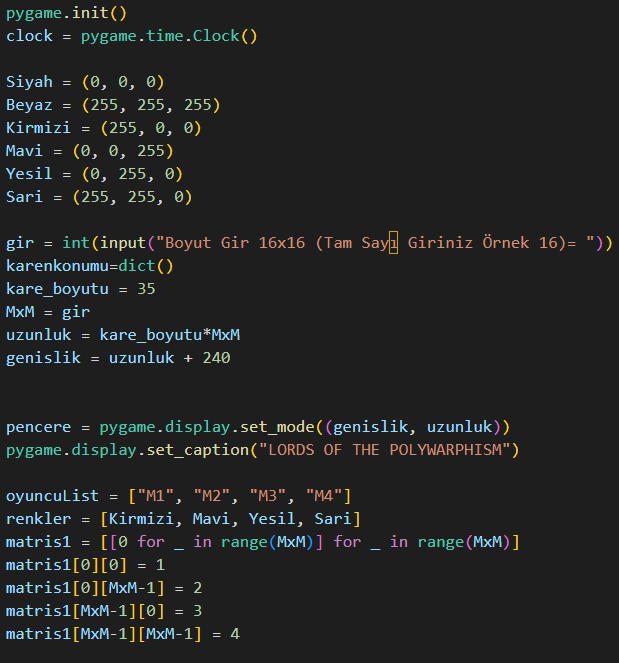
metin, ekran görüntüsü, diyagram, kare içeren bir resim

Açıklama otomatik olarak oluşturuldu

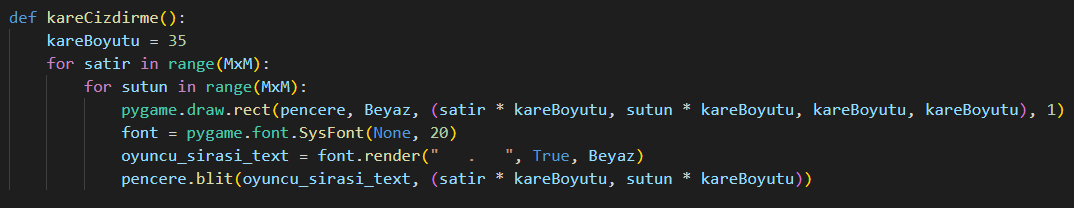
* 1. **Kullanıcı arayüzü tasarımı**



Proje Gerçekleştirlirken Kullanılan Kütüphaneler



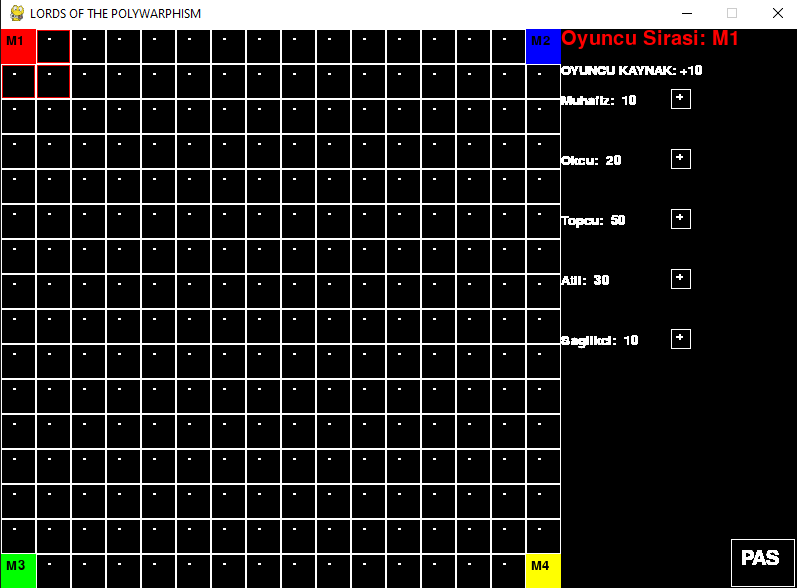
Pencere Oluşturma, Matris Tasarımı , Pencere İsmi Belirleme Oyuncu İsmi ve Renkleri Belirleme



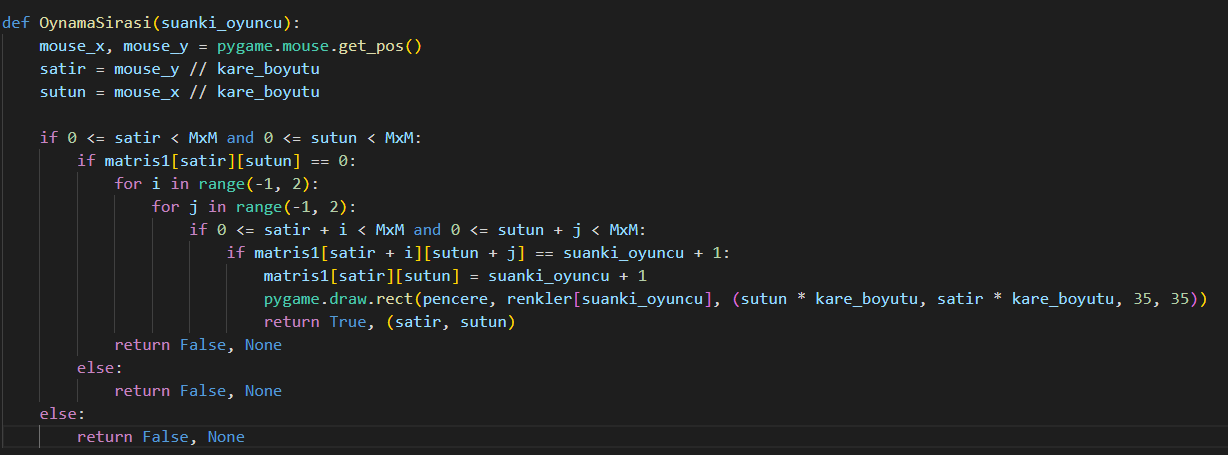
Kullanıcı Tarafından Girilen Boyuta Göre MxM Kare Dünya Oluşturma



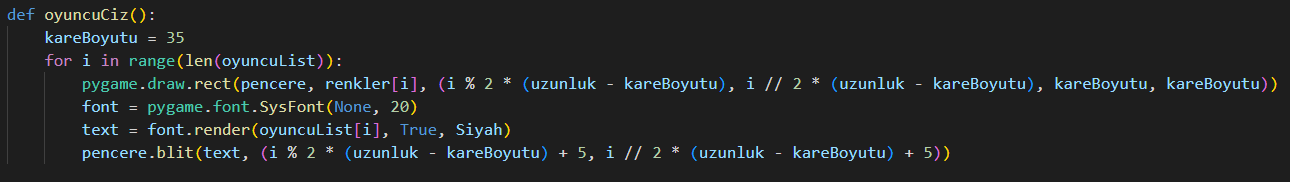
Pygame kütüphanesinin draw.rect ve blit fonksiyonları sayesinde arayüz tasarlama kullanıcı tarafından belirlenen uzunluk ölçütlerine göre değişkenlik gösterir.



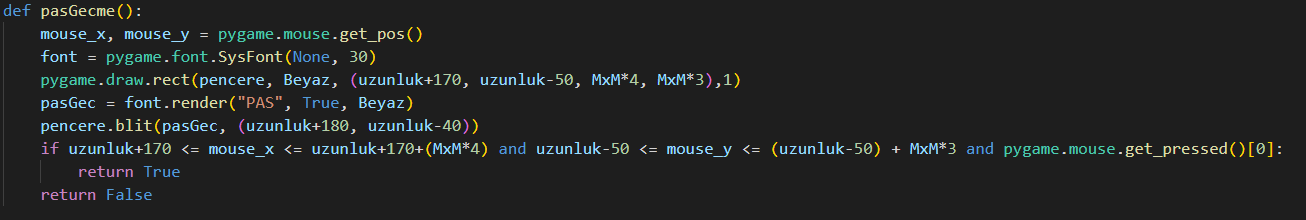
Arayüz



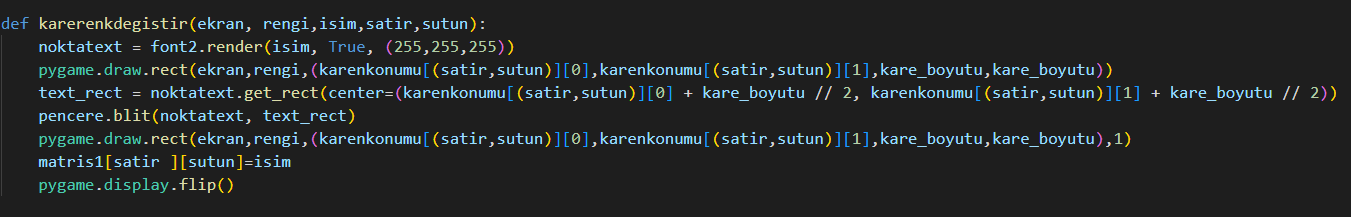
Kullanıcıların Oynama Sırasına Göre İşlem Yapmasına Yarayan OynamaSırası Fonksiyonu



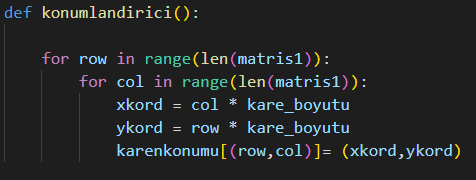
OyuncuCizdirme Fonksiyonu



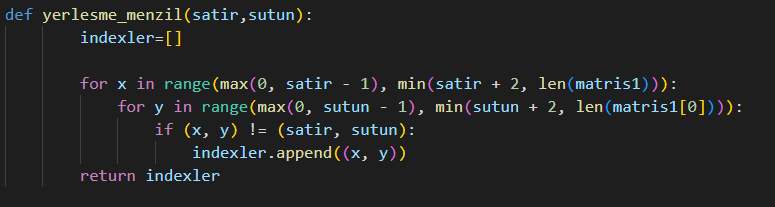
Hamle Yapmak İstemeyen Kullancıların Kullanması İçin Pas Geçme Fonksiyonu



Kullanıcı Tarafından Seçilen Karelerin Renklerini Değiştirme Fonksiyonu



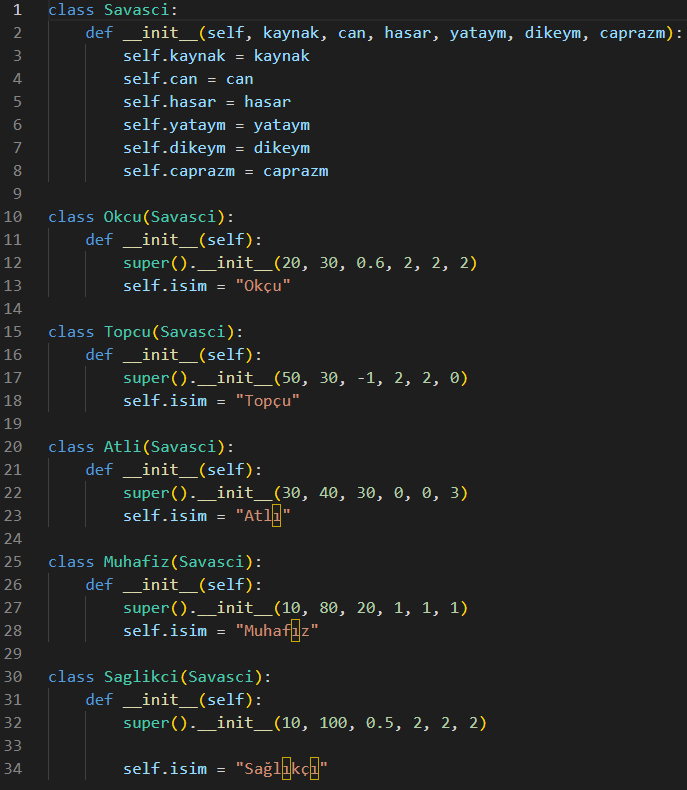
Konum Belirleme Fonksiyonu



Menzil Tespit Etme Fonksiyonu



Simülasyonun Çalışma Mekanizması



Polymorphism ile sınıf oluşturma

1. **UYGULAMA**
   1. **Kodlanan bileşenlerin açıklamaları**

* “Savasci” sınıfı, genel savaşçı özelliklerini ve niteliklerini tanımlar.
* Okcu, Topcu, Atli, Muhafiz ve Saglikci sınıfları, Savasci sınıfından türetilmiş olup, her biri belirli özelliklere ve niteliklere sahip farklı bir savaşçı türünü temsil eder.
* matrisPanel(), konumBelirleme(), karelereAyırma(), kareRenkDegistirme(), oyuncuKontrol(), oyuncuSırası(), ArayüzTasarlama(), AnlıkKoordinat(), oynamaSırasıBelirleme(), savasciYerlestirme(), ArayüzTasarlamaGosterge() ve mainGame() gibi işlevler, oyunun farklı yönlerini kontrol eder ve kullanıcı arayüzüyle etkileşime girer.
* pasGecme() işlevi, oyuncunun bir hamle yapmadan geçmesini sağlar.
* Ana işlev olan mainGame(), oyunun ana döngüsünü ve oyun mekaniğini yönetir. Oyuncuların sırayla hamle yapmasını, hamle geçişlerini, taşın yerleştirilmesini ve arayüzün güncellenmesini sağlar. Bu işlev, bir Pygame penceresi oluşturur ve oyunun çalışmasını sağlar. Bu kod parçalarını adım adım inceleyerek, oyunun işlevselliğini ve nasıl çalıştığını daha iyi anlayabilirsiniz.
  1. **Görev dağılımı**
* Bileşenlerinin tasarım ve geliştirme aşamalarında Kütüphanelerin öğrenilmesi , Algoritma hazırlanması ve deneme yöntemleri ortak gerçekleştirilmiştir.
* Raporun hazırlanması süreçlerinde ortak çalışma yapılmıştır.
  1. **Karşılaşılan zorluklar ve çözüm yöntemleri**
* Alınan tıklama bilgilerini matrise geçirmek.
* Matris Paneline işlenen verilerin dinamik kalması
* Oyuncu Sırası Kontrolü
* Oyuncu Sırası İle Entegre Pas İşlemi
* Kullanıcı Tanımlı Matris Sistemi Oluşturma
* Matris Sisteminin Arayüz ile Entegre Çalışması
* Kullanıcılara Özel Renklerin Tanımlanması
* Tanımlanan Renklere Göre Kare Boyama İşlemi
* Oyuncu Kaynak Kontrolü
* Savaşcı Menzil Kontrolü
* Her Kullanıcının Kendi Belirlenen Bölgelerine Göre Kare Boyaması
* Her Turda 2 Adet Kare Boyama Hakkı
* Arayüz Tasarımı
* Yukarıda Belirtilen Zorluklar Proje İçerisinde Belirtilen Fonksiyonlar ve Sınıflar Yardımı İle Giderilmiştir.
  1. **Proje isterlerine göre eksik yönler**
* Savaşma ve Bölge İşgal Etme Konusunda Eksiklikler

1. **TEST VE DOĞRULAMA**
   1. **Yazılımın test süreci**

* Proje kapsamında Yazılan Tüm Sınıflar ve fonksiyonlar test edilmiştir
  1. **Yazılımın doğrulanması**
* Proje kapsamında Testi Yapılan Sınıflar ve Fonksiyonlar üzerinde tespit edilen anormallikler giderilmiştir.