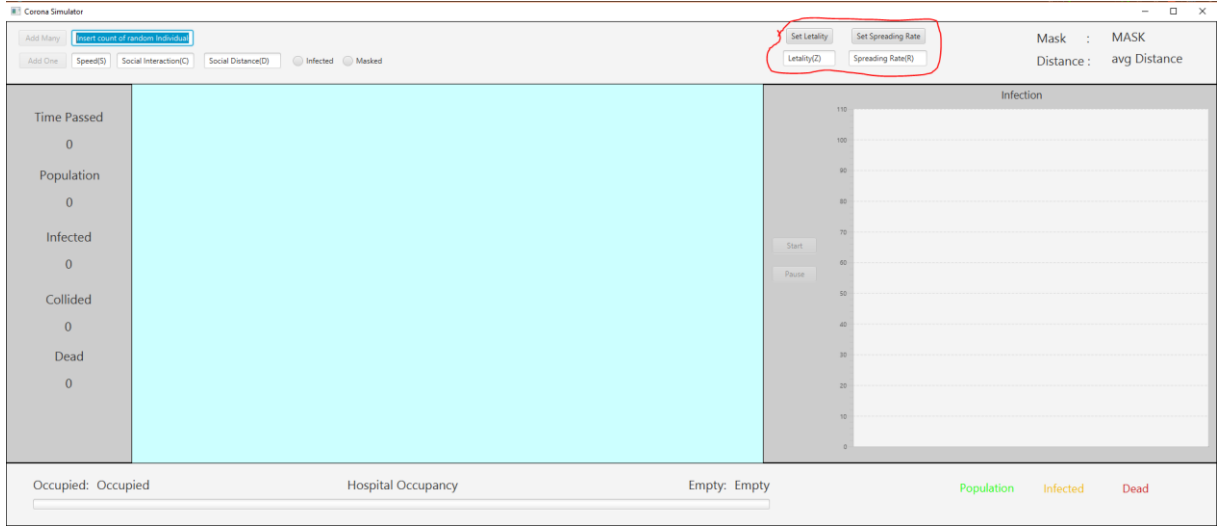


# Gebze Technical University Computer Engineering

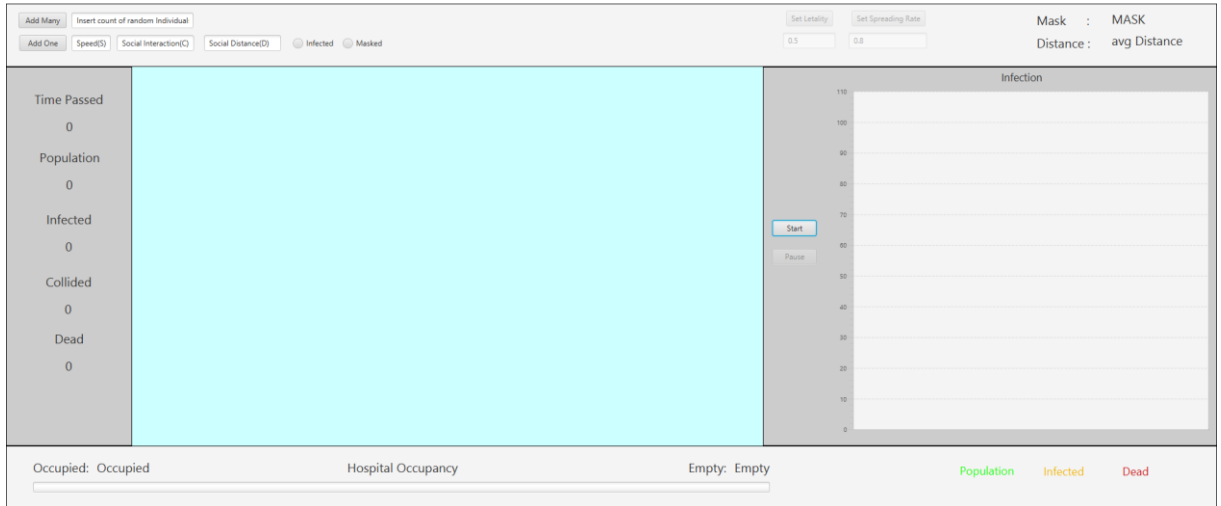
## CSE 443 Final Assignment Report

Barış Şahin  
151044016

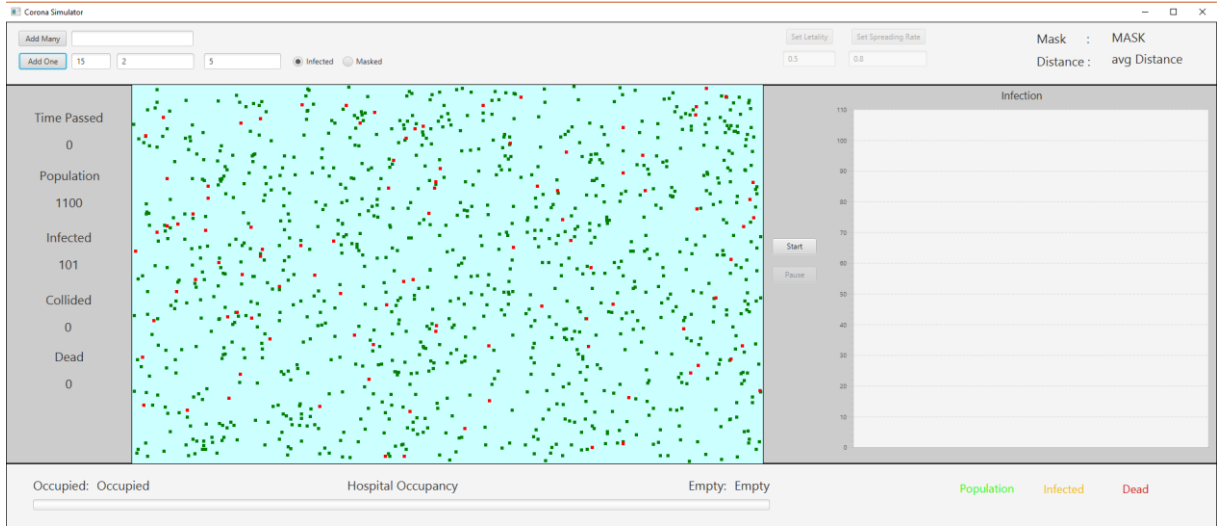
# Kullanım



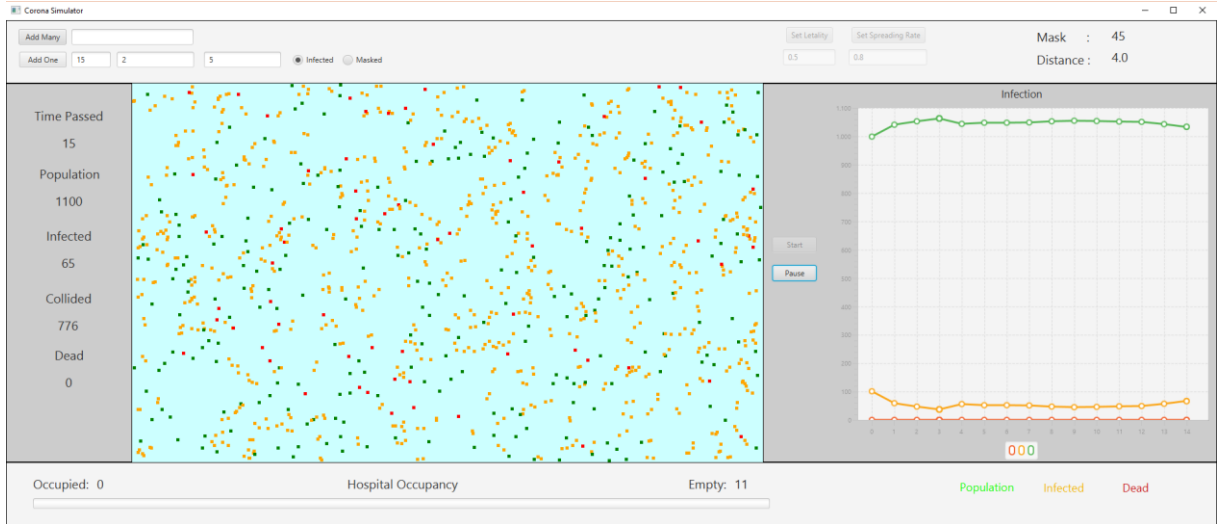
Öncelikle Z değeri ve R değeri girilmelidir. Z değeri girilmeden deathTimer hesaplanamayacağı için bu iki değer girilene kadar simülasyonun başlamasına veya insan eklemeye izin verilmez.



Bu değerler girildikten sonra insan ekleme butonları ve başlat butonu aktif olur.



Gerekli barlar doldurularak toplu veya tek tek insan üretilebilir. Hiç insan olmadan da simülasyon başlatılabilir fakat maske takanların yüzdesi, ortalama sosyal mesafe ve hastahenenin boyutu simülasyonun başladığı zamanki popülasyona göre belirlenir.

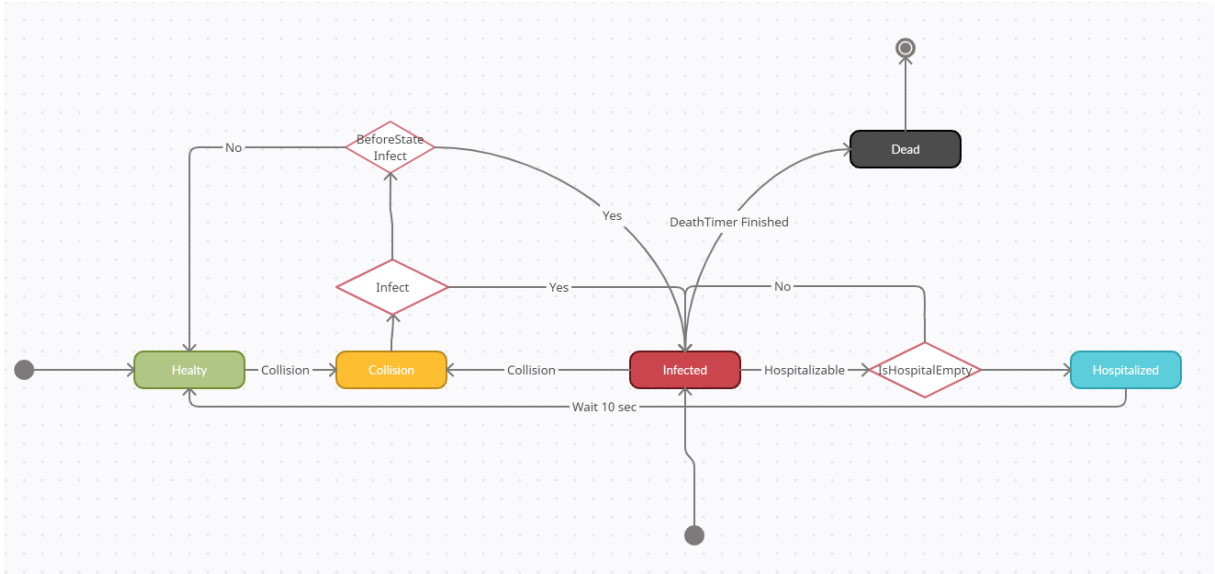


Durdur tuşu sadece simülasyon aktifken aktif olur.

# Kullanılan Örüntüler

## Durum Tasarım Örüntüsü

### Health State Diagram



Bu simülasyonda insanın davranışları içinde bulunduğu duruma göre değişmektedir. Bu durumlarda haritadaki şekli ve hareket etmeden önce kontrol etmesi gereken şeyleri daha düzenli tutmak adına insanın sağlık durumlarını State Pattern deki durumlar olarak aldım.

Ayırdığım durumlar olarak:

**HealthyState:** Bu durumda insan normal bir şekilde hiçbir kontrol yapmadan hareket edebiliyor ve kendini yeşil olarak çizdiriyor.

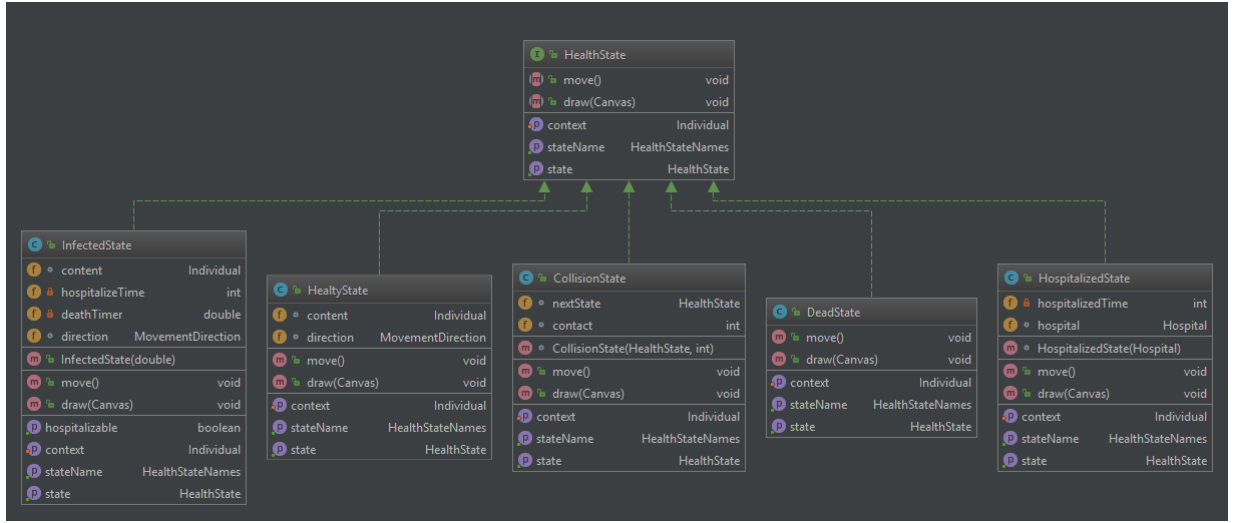
**InfectedState:** Bu durumdayken insan hareket etmeden önce bu duruma geçerken atanan ölüm zamanını kontrol ediyor. Eğer ölüm zamanı geldiyse kendini DeadState'e geçiriyor. Aynı zamanda bu durumdaki bir insan bu durumdaki 25. Saniyesinde bu duruma özgü hastahaneye kaldırılabilir bayrağını doğru olarak belirliyor. Böylece ana döngüdeki fonksiyonlar tarafından arabulucu sınıfa yönlendirilip uygun bir hastahaneye yatırılabilir. Kendilerini kırmızı olarak çizdiriyorlar.

**HospitalizedState:** İnsan bu durumda hareket edemiyor ve kendini çizdiremiyor. İnsan hareket fonksiyonunda taburcu olmayı bekliyor. Taburcu olma zamanı geldiğinde arabulucu sınıf aracılığı ile hastahaneden taburcu oluyor.

**CollideState:** İki insanın ilişkisi arabulucu sınıf tarafından karşılaşıyor olarak tespit edilirse karşılaşan insanlar bu duruma düşüyor. Arabulucu hesaplamalarının sonucunda karşılaşma bittiğinde hangi duruma geçilmesi gerektiğini ve karşılaşmanın ne zaman biteceğini bu duruma geçerken parametre olarak veriyor. Bu durumdaki insanlar hareket etmiyor ve karşılaşma zamanının bitmesini bekliyorlar. Kendilerini turuncu olarak çizdiriyorlar.

**DeadState:** Sadece kendini ve ismini döndürebilen ölü durumu. Bu durumdaki insanlar hareket edemez ve kendini çizdiremez. Bu son durumdur.

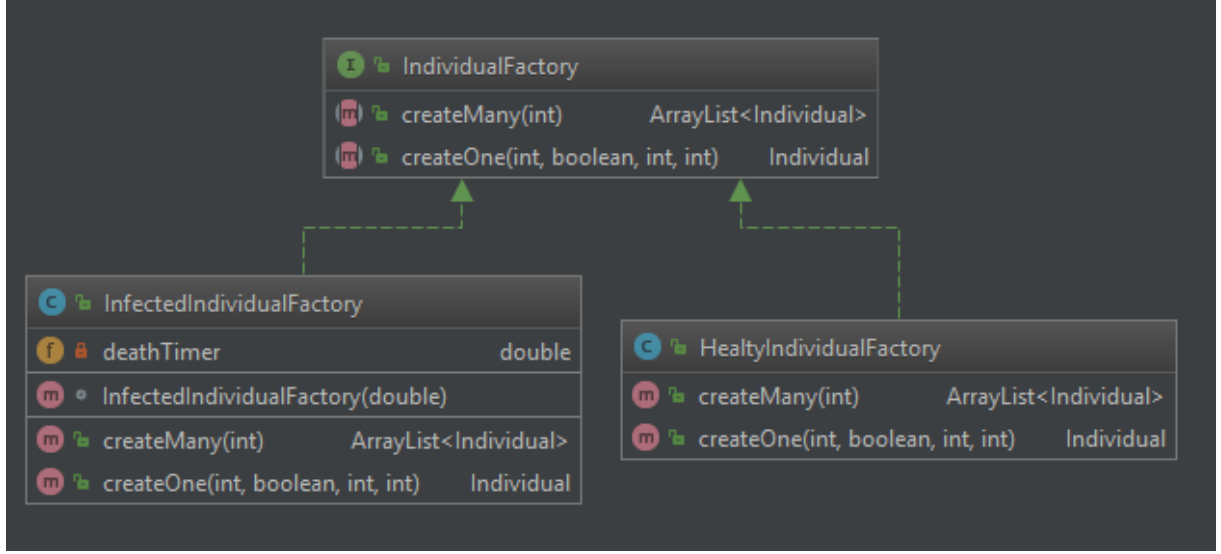
Bu durumların bana faydaları move ve render fonksiyonları insanın durumuna göre değişiyor olması. Ayrıca olaylara göre durumlar arasındaki geçiş bunu yönetmemi kolaylaştırıyor.



## Arabulucu Tasarım Örüntüsü

Arabulucu tasarım örüntüsünü çarpışma durumlarını ve hastahane ile insan arasındaki ilişkiyi yönetmek için kullandım. Bütün insanları arabulucu sınıfın içinde tutmak yerine arabulucu sınıfın fonksiyonlarını statik yaparak bütün ilgili objelerden erişilebilmesini sağladım. Bir çarpışma veya hastahane ile insan arasındaki bir etkileşim durumunda ilişkide bulunacak kişiler arabulucu sınıfın fonksiyonlarına parametre olarak veriliyor. Arabulucu hesapları yaptıktan sonra insanların durumlarını olması gerektiği gibi ayarlıyor.

## Fabrika Tasarım Örüntüsü



Farklı durumlardaki insanları çoklu veya tekli ürebilmek için fabrika tasarım örüntüsünü kullandım.

## Testler ve Grafiker

Testleri ve testlere ait grafikleri Test klasörü içinde bulabilirsiniz. Genel olarak gözlemlediğim Hastalıklı insan sayısındaki değişim çok değilken popülasyon exponansiyel olarak azalırken ölümler de aynı oranda artmaktadır.