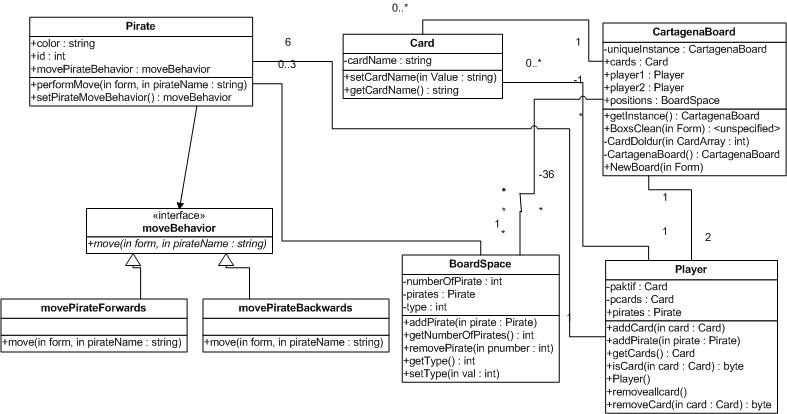
**CARTAGENA GAME**

**Hüseyin ÖRŞAN**

**Compiler: Visual Studio 2010 C#**

**Uml Diagram**

****

**CartagenaBoard Class:**

Bu class oyunun boardunu veüzerinde random oluşacak kartları forma çizer. Form üzerinde oyun itemleri ile ilgili objelerin hiçbiri yoktur. Bu sınıf tüm itemleri hangi formdan çağrılıyorsa o forma çizer. Oyun çalıştırıldığında New Game butonuna basıldığında 2 player objesi oluşturulur. Bu playerlara başlangıçta 6 şar adet kart dağıtılır. Bu class da ayrıca **singeton pattern** kullanılarak classın ınstance ı bir kez oluşturulur ve her NEW Game butonuna basıldığında kartlar tekrar dağıtılır fakat singelton pattern den dolayı ınstance aynıdır.

CartagenaBoard Class ı 36 adet boardSpace sınıfı tipinde pozisyona sahiptir. Bu pozisyonlar rastgele dağtılmıştır. Pozisyonlar pistol, şapka, kılıç, şişe, anahtar ve anahtar tipinde her birinden 6 şar adet kartlardan oluştuğundan bu sınıf Card sınıfının özelliklerini kullanılır. Boxclean ve NewBoard methodları da boardu hazırlamak için oluşturulmuştur.

**Player Class:**

Player sınıfı 6 adet pirate e ve başlangıçta 6 adet karda sahiptir. Pirate sınıfından 6 instance her bir player için oluşturulur. Sınıfa sonradan yeni kart eklenebilir ve kart remove edilebilir. Bu sebeple Card sınıfını kullanılır ve kartlar dinamik array de tutulur. Başlangıçta 6 adet karta sahipken sonrasında kart sayısı artabilir. Player Pirate lerini Pirate sınıfı ile move eder.

**Player Class ın methodları:**

**addCArd methodu :** Bu methodla card sınıfı kullanılarak yeni bir kart eklenir.

**addPirate methodu:** Bu methodla 6 adet pirate player a eklenir.

**addPirate methodu:** Bu methodla 6 adet pirate player a eklenir.

**getCards methodu:** Bu methodla player ın sahip olduğu tüm kartlar öğrenilebilir.

**İscard methodu:** Bu methodla player ın ilgili karta sahip olup olmadığı öğrenilir. (Form üzerinde 6 farklı itemin resmi her daim mevcuttur yanlarındaki kutularda player in bu itemlerden ne kadarına sahip olduğu kullanıcıya gösterilmesine rağmen bu karta sahip olup olmadığını sorgulayarak oyun kurallarını koruyoruz.)

**removeallCArd methodu:** Player ın kartlarını dinamik arrayda tuttuğumuzdan her NEW Game butonuna bastığımızda (singelton pattern kullanıldığı için) player ın tüm kartlarını remove ediyoruz.

**removeallCArd methodu:** Oyuncu kart seçip piyonları (pirate) hareket ettirdiğinde elindeki kartı kullanmış olduğundan o kartı player den kaldırıyoruz.

**BoardSpace Class:**

36 adet yer için her birine max 3 er pirate alabilir. Her bir pirate Pirate Sınıfından türetilir.

**BoardSpace Class ın methodları:**

**addPirate methodu:** Bu methodla boardun ilgili pozisyonuna pirate eklenir. Oyun kuralları gereği her bir pozisyonda 3 adet pirate olabilir.

**getNumberOfPirates methodu :** Bu methodla boardun ilgili pozsiyonun kaç adet pirate e sahip olduğu öğrenilir. 3 den fazla pirate olamayacağı için bu methodla kontrol sağlanır.

**removePirate methodu :** Bu methodla boardun ilgili pozsiyonundan ilgili pirate remove edilir.

**gettype methodu :** Pozisyonun hangi item olduğu bilgisini döner.

**setType methodu :** Pozisyona ilgili itemi set etmek için kullanılır.

**Card Class:**

İlgili kartın ismi set edilir ve get edilir. Player sınıfı ve CartagenaBoard sınıfı bu sınıftan kartlarını oluşturur.

**Pirate Class: (Strateji Pattern)**

performMove methodu ile behavior class a delegate eder. Player ın piratelerini forward ya da backward yapmak için kulanılır. Hangi yönde hareket edilecekse setPirateMoveBehavior methodu ile behavior değiştirilebilir.

**Encapsulated moveBehavior:**

**İnterface moveBehavior:** movePirateBackwards ve movePirateForwards class ları tarafından implement edilir ve move methodu bacwards ve forwards methodları tarafından replace edilir.