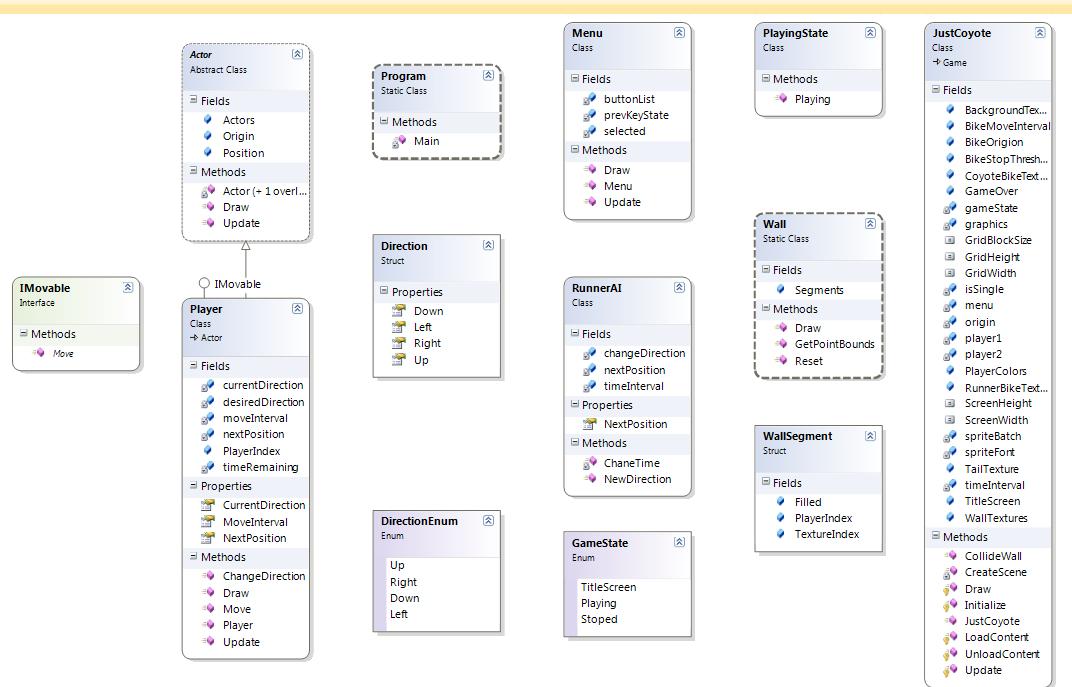
Team "Wile E. Coyote"

Целта на отбора е разработка на игра, обектно ориентирана. Тима се казва Wile E. Coyote и затова самата задача е сързана с този анимационен филм ,Wile се гони с едно пиле . При тази игра пилето и койотът оставят следи след себе си и целта е да не са блъснеш в твоята следа или в тази на другия.Играта има опция да се играе от един или двама играча.

Самата игра е от 8 класа: Actor.cs ; Menu.cs; Player.cs; PlayingState.cs; RunnerAI.cs; Wall.cs; JustCoyote.cs; Program.cs. И един Интервейс IMovable.cs. И две структури WallSegment.cs и Direction.cs. Две енумерации DirectionEnum.cs и GameState.cs .

Технологиите използвани за направата на играта са : Visual Studio 2012 , XNA studio. URL на проекта е <https://coyoteteam.visualstudio.com/DefaultCollection/JustCoyote>. Клас диаграмата на играта е :



От класа Program.cs се стартира самата игра. Втория клас JustCoyote.cs , той е наследник на Microsoft.Xna.Framework.Game, определя размера на игралното поле , интервала през които ще се движат играчите по полето, визията на играчите, фона на играта. Също така , след като потребителя е избрал кaкъв тип игра иска да играе , със цикъл switch се подразбира какъв тип игра да започне: Single или MultiPlayer тип.

Структурата WallSegment е съставена от три променливи Filled,TextureIndex и PlayerIndex, Тя е създадена, за да може от нейните променливи да се определи , кога ще свърши играта.

Класът Wall определя дали опашката , която оставя играча след себе си ще се пресече от самия себе си или от другия играч , при което играта ще свърши. Също така класа използва структурата WallSegment , от който се използват нейните променливи като масив. Класа има три метода Draw, GetPointBounds и Reset. Първия Draw определя как да се запълва полето със следите на на играчите и цвета на следата. Втория GetPointBounds , определя как да изглежда следата и какви размери да има, по x и по y. Третия Reset , връща размера на полето.

Класа Actor.cs абстрактен клас, определя вектор , който е позицията на играча , и също така има два метода , виртуални ,единия от които рисува следата, а другия ъпдейтва състоянието на играча.

Структурата Direction.cs , има четири свойства - вектора с две компоненти по х и по у . Те определят посоката на движение на играчите. Посоките на движение се избират от енумерацията DirectionEnum.cs , в която са зададени Up, Down, Left,Right.

Другата енумерация в проекта GameState.cs определя различните дисплеи на играта :Начален , в процес на игра , и стопиран.

Интерфейса IMovable , декларира метода Move.

В класа Menu.cs се избира какъв тип игра да се играе, или пък да се излезе от самата игра. Опците се изобразяват чрез List.Начина на избиране е чрез метода Update, той прихваща дали бутоните са натиснат надолу или нагоре. Чрез метода Draw се оцветява избора на играча.

Класа Player наследява класа Actor и интерфейса IMovable , като има няколко метода , които служат за движението , промяната на посоката и рисуването на следата на играча

Класа PlayingState има един метод Playing , който да определя с кои клавиши да се извършва движението при един и много играчи, като всичко това е в цикъл fo r , който обикаля през броя на играчите.

Класа RunnerAI определя посоката на движение на пилето , когато е в режим Single player , през определен интервал .