Class Enemy   </>

Namespace: [Entity](#)

Assembly: Spicy_Invaders.dll

Enemy Class which is a SmartEntity

```
public class Enemy : SmartEntity
```

Inheritance

[object](#) ← [MovableEntity](#) ← [SmartEntity](#) ← Enemy

Inherited Members

[SmartEntity.Weapon](#) , [SmartEntity.HealthPoints](#) , [SmartEntity.FaceDirection](#) ,
[SmartEntity.ShootXPos](#) , [SmartEntity.ShootYPos](#) , [SmartEntity.EntityWidth](#) , [SmartEntity.IsAlive](#) ,
[SmartEntity.IsHit](#) , [SmartEntity.Shoot()](#) , [MovableEntity.TravelDirection](#) , [MovableEntity.Position](#) ,
[MovableEntity.Velocity](#) , [MovableEntity.Move(Direction)](#) , [object.Equals(object)](#) ,
[object.Equals(object, object)](#) , [object.GetHashCode()](#) , [object.GetType()](#) ,
[object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) , [object.ToString()](#)

Constructors

Enemy(int, int, EnemyType, Direction)   </>

Constructor for enemies which sets the different attributes based on the enemy type.

```
public Enemy(int x, int y, EnemyType type, Direction direction = Direction.Right)
```

Parameters

x [int](#)

current x pos

y [int](#)

current y pos

type [EnemyType](#)

the type of enemy

direction [Direction](#)

current travling direction

Properties

CanFire   </>

if the enemy can fire projectiles

```
public bool CanFire { get; set; }
```

Property Value

[bool](#)⧉

ExplosionLevel   </>

What explosion level it's at.

```
public int ExplosionLevel { get; set; }
```

Property Value

[int](#)⧉

Points   </>

The amount of points a enemy gives when destoryed-

```
public int Points { get; set; }
```

Property Value

[int](#)⧉

Type   </>

The enemy's type

```
public EnemyType Type { get; }
```

Property Value

[EnemyType](#)

Methods

Hit(Projectile)   </>

override for Hit method which lowers enemy's healthpoints by the projectile type's damage. If health points are zero than enemy is dead and starts to explode.

```
public override void Hit(Projectile projectile)
```

Parameters

projectile [Projectile](#)