

# Class MovableEntity

&lt;/&gt;

Namespace: [Entity](#).

Assembly: Spicy\_Invaders.dll

MovableEntity class, which is any object in the game which moves.

```
public class MovableEntity
```








## Inheritance

[object](#)  ← MovableEntity

## Derived

[Drop](#), [Projectile](#), [SmartEntity](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#) 

## Constructors

### MovableEntity()

&lt;/&gt;

Constructor.

```
protected MovableEntity()
```

## Properties

### Position

&lt;/&gt;

The entity's current position

```
public Vector Position { get; set; }
```

### Property Value

[Vector](#)

## TravelDirection

&lt;/&gt;

The direction in which the entity is moving.

```
public Direction TravelDirection { get; set; }
```

Property Value

[Direction](#)

## Velocity

&lt;/&gt;

the speed by which the entity moves.

```
public Vector Velocity { get; set; }
```

Property Value

[Vector](#)

## Methods

### Move(Direction)

&lt;/&gt;

Move Method which updates the entity's position based on it's move speed (velocity), it's current position, and travel direction.

```
public void Move(Direction directionToGo = Direction.None)
```

Parameters

**directionToGo** [Direction](#)

Direction param if the entity to move it in a different direction than it's current travel direction