

Class Alien

Namespace: [Model](#)

Assembly: Model.dll

La classe Alien contient les caractéristiques de bases de l'alien et contient les actions qu'il peut effectuer

```
public class Alien
```

Inheritance

[object](#) ← Alien

Derived

[AlienBoss1](#)

Inherited Members

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

Constructors

Alien(int, int, int)

Constructor de l'invader pour définir ces paramètres par défaut

```
public Alien(int x, int y, int hpDefault)
```

Parameters

x [int](#)

y [int](#)

hpDefault [int](#)

Fields

AlienisAlife

```
public bool AlienisAlife
```

Field Value

[bool](#)↗

BoomNotArrived

```
public bool BoomNotArrived
```

Field Value

[bool](#)↗

hpDefault

```
public int hpDefault
```

Field Value

[int](#)↗

invaderDirection

```
public bool invaderDirection
```

Field Value

[bool](#)↗

width

```
public const int width = 150
```

Field Value

[int](#)

X

```
public int x
```

Field Value

[int](#)

y

```
public int y
```

Field Value

[int](#)

Methods

TakeDamage(missileJoueur)

On inflige des dégats a l'alien si il est toucher par le missile du joueur

```
public virtual void TakeDamage(missileJoueur Missilejoueur)
```

Parameters

Missilejoueur [missileJoueur](#)

chargementAlien(missileAlien)

On ajoute nos missile dans l'alien

```
public virtual void chargementAlien(missileAlien missileDefault)
```

Parameters

`missileDefault` [missileAlien](#)

dropMissileAlien()

On lance nos missiles et on les enlève de l'alien

```
public missileAlien dropMissileAlien()
```

Returns

[missileAlien](#)

moveLeft()

On bouge à gauche de 1, et dès qu'on arrive à la fin de la fenêtre on descend de 2

```
public virtual void moveLeft()
```

moveRight()

On bouge à droite de 1, et dès qu'on arrive à la fin de la fenêtre on descend de 2

```
public virtual void moveRight()
```

Class AlienBoss1

Namespace: [Model](#)








Assembly: Model.dll

```
public class AlienBoss1 : Alien
```

Inheritance

[object](#)  ← [Alien](#) ← AlienBoss1

Inherited Members

[Alien.width](#), [Alien.x](#), [Alien.y](#), [Alien.hpDefault](#), [Alien.invaderDirection](#), [Alien.AlienIsAlive](#), [Alien.BoomNotArrived](#), [Alien.chargementAlien\(missileAlien\)](#), [Alien.dropMissileAlien\(\)](#), [Alien.TakeDamage\(missileJoueur\)](#), [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Constructors

AlienBoss1(int, int, int)

```
public AlienBoss1(int x, int y, int hpDefault)
```

Parameters

x [int](#) 

y [int](#) 

hpDefault [int](#) 

Methods

moveLeft()

On bouge a gauche de 1, et dés qu'on arrive a la fin de la fenetre on change de sens

```
public override void moveLeft()
```

moveRight()

On bouge a droite de 1, et dés qu'on arrive a la fin de la fenetre on chnage de sens

```
public override void moveRight()
```

Class Missile

Namespace: [Model](#)

Assembly: Model.dll

La classe Missile contient les caractéristiques de bases qui vont être héritées aux classes missile alien et missile joueur

```
public class Missile
```

Inheritance

[object](#) ← Missile

Derived

[missileAlien](#), [missileJoueur](#)

Inherited Members

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

Fields

damage

```
public int damage
```

Field Value

[int](#)

missileIsLaunched

```
public bool missileIsLaunched
```

Field Value

[bool](#)

X

public int x

Field Value

[int](#)

y

public int y

Field Value

[int](#)

Class missileAlien

Namespace: [Model](#)

Assembly: Model.dll








La classe Missile alien contient les caractéristiques de bases du missile alien et les actions qu'il peut effectuer

```
public class missileAlien : Missile
```

Inheritance

[object](#)  ← [Missile](#) ← missileAlien

Inherited Members

[Missile.damage](#) , [Missile.missileIsLaunched](#) , [Missile.x](#) , [Missile.y](#) , [object.Equals\(object\)](#)  ,
[object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Constructors

missileAlien(int)

Constructor du missile avec ces valeurs par défaut

```
public missileAlien(int damage)
```

Parameters

damage [int](#) 

Fields

height

```
public int height
```

Field Value

[int](#) 

Methods

UpdateMisille()

On déplace notre missile

```
public void UpdateMisille()
```

Class Hpcoeur1


Namespace: [Model](#)

Assembly: Model.dll








La classe Hpcoeur contient la position qui va etre utiliser pour l'affichage des coeurs dans le jeu

```
public class Hpcoeur1
```

Inheritance

[object](#)  ← Hpcoeur1

Inherited Members

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

Constructors

Hpcoeur1(int, int)

Constructor d'un hp pour définir ces paramètres par défaut

```
public Hpcoeur1(int x, int y)
```

Parameters

x [int](#) 

y [int](#) 

Fields

x

```
public int x
```

Field Value

[int](#) 

y

```
public int y
```

Field Value

[int](#)

Class missileJoueur

Namespace: [Model](#)

Assembly: Model.dll








La classe Missile joueur contient les caractéristiques de bases du missile joueur et les actions qu'il peut effectuer

```
public class missileJoueur : Missile
```

Inheritance

[object](#)  ← [Missile](#) ← missileJoueur

Inherited Members

[Missile.damage](#) , [Missile.missileIsLaunched](#) , [Missile.x](#) , [Missile.y](#) , [object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Constructors

missileJoueur(int)

Constructor du missile avec ces valeurs par défaut

```
public missileJoueur(int damage)
```

Parameters

damage [int](#) 

Fields

missileIsTouched

```
public bool missileIsTouched
```

Field Value

[bool](#) 

Methods

UpdateMisille()

On déplace notre missile

```
public void UpdateMisille()
```

Class Player

Namespace: [Model](#)

Assembly: Model.dll








La classe Player contient les caractéristiques de bases du joueur et contient les actions qu'il peut effectuer

```
public class Player
```

Inheritance

[object](#)  ← Player

Inherited Members

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

Constructors

Player(int, int, int)

Constructor du joueur pour définir ces paramètres par défaut

```
public Player(int x, int y, int hpDefault)
```

Parameters

x [int](#) 

y [int](#) 

hpDefault [int](#) 

Fields

CoeurRemove

```
public bool CoeurRemove
```

Field Value

[bool](#)

PlayerisAlife

```
public bool PlayerisAlife
```

Field Value

[bool](#)

hpDefault

```
public int hpDefault
```

Field Value

[int](#)

missileIsTouched

```
public bool missileIsTouched
```

Field Value

[bool](#)

X

```
public int x
```

Field Value

[int](#)

y

```
public int y
```

Field Value

[int](#)

Methods

TakeDamage(missileAlien)

On inflige des dégâts a au joueur si il est toucher par le missile de l'alien

```
public void TakeDamage(missileAlien MissileAlien)
```

Parameters

MissileAlien [missileAlien](#)

chargement(missileJoueur)

On ajoute nos missile dans le joueur

```
public void chargement(missileJoueur missileDefault)
```

Parameters

missileDefault [missileJoueur](#)

dropMissile()

On lance nos missiles et on les enlèvent du joueur

```
public Missile dropMissile()
```

Returns

[Missile](#)

moveDown()

```
public void moveDown()
```

moveLeft()

On décale à gauche de 1 l'axe x du joueur

```
public void moveLeft()
```

moveRight()

On décale à droite de 1 l'axe x du joueur

```
public void moveRight()
```

moveUp()

```
public void moveUp()
```