

# Namespace SpicyInvaders

## Classes

[Alien](#)

[Bullet](#)

[DataBaseConnect](#)

[ObjectBase](#)

[Player](#)

[Score](#)

Clas SCORE

# Class MainCode

Namespace: [SpicyInvaders](#)








Assembly: SpicyInvaders.dll

```
public class MainCode
```

## Inheritance

[object](#)  ← MainCode

## Inherited Members

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

## Methods

### Main(string[])

```
public static void Main(string[] args)
```

## Parameters

args [string](#)  []

# Class GameEngine

Namespace: [SpicyInvaders](#)

Assembly: SpicyInvaders.dll

```
public class GameEngine
```



## Inheritance

[object](#) ← GameEngine

## Inherited Members

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

## Constructors

### GameEngine()

```
public GameEngine()
```

## Methods

### BulletMovement(List<Bullet>)

Methodo qui gère le mouvement de tous les bullets à l'écran.

```
public static void BulletMovement(List<Bullet> bullets)
```

## Parameters

**bullets** [List](#) <[Bullet](#)>

### BulletsDraw(List<Bullet>)

Methodo pour afficher tous les bullets dans le jeu

```
public static void BulletsDraw(List<Bullet> bullets)
```

## Parameters

bullets [List](#) <[Bullet](#)>

## CollisionSystem(List<Alien>, List<Bullet>, Player, Score)

Methodo qui contrôle les collisions du jeu et sa réponse.

```
public static void CollisionSystem(List<Alien> enemies, List<Bullet> bullets, Player player, Score score)
```

## Parameters

enemies [List](#) <[Alien](#)>

bullets [List](#) <[Bullet](#)>

player [Player](#)

score [Score](#)

## ControlMenu()

Methodo pour afficher le menu de controles du jeu

```
public static void ControlMenu()
```

## EndGame(Player, Score, bool)

Methodo qui affiche le menu de fin de partie avec les données de la partie.

```
public static void EndGame(Player player, Score score, bool savePoints)
```

## Parameters

player [Player](#)

score [Score](#)

savePoints [bool](#)

## EnemiesAttack(List<Alien>, List<Bullet>)

Methodo qui gère l'attaque de tous les enemies avec des bullets

```
public static void EnemiesAttack(List<Alien> enemies, List<Bullet> bullets)
```

### Parameters

enemies [List](#) <[Alien](#)>

bullets [List](#) <[Bullet](#)>

## EnemiesDraw(List<Alien>)

Methodo pour afficher tous les ennemis dans le jeu

```
public static void EnemiesDraw(List<Alien> enemies)
```

### Parameters

enemies [List](#) <[Alien](#)>

## EnemiesMovement(List<Alien>, int, int)

Methodo qui gère le mouvement de tous les ennemis à l'écran.

///

```
public static void EnemiesMovement(List<Alien> enemies, int limitMapLeft,  
int limitMapRight)
```

### Parameters

enemies [List](#) <[Alien](#)>

limitMapLeft [int](#)

limitMapRight [int](#)

## HighScoreMenu()

Methodo pour afficher le menu de Highscore du jeu

```
public static void HighScoreMenu()
```

## HudGameDraw(Player, Score)

Methodo pour afficher l'interface du jeu

```
public static void HudGameDraw(Player player, Score score)
```

Parameters

player [Player](#)

score [Score](#)

## PlayerControll(Player, int, int, List<Bullet>)

Methodo qui gère le contrôle du joueur avec son navire en fonction des touches.

```
public static void PlayerControll(Player player, int limitMapRight, int limitMapLeft, List<Bullet> bullets)
```

Parameters

player [Player](#)

limitMapRight [int](#)

limitMapLeft [int](#)

bullets [List](#) <[Bullet](#)>

## StartGame()

Initialisation du jeu avec les objets et l'interface

```
public static void StartGame()
```

# Title()

Menu principal du jeu

```
public static void Title()
```

## Update(Player, List<Alien>, List<Bullet>, Score)

Methodo qui met constamment à jour le jeu pour son gameplay.

```
public static void Update(Player player, List<Alien> enemies, List<Bullet> bullets,  
Score score)
```

### Parameters

player [Player](#)

enemies [List](#) <[Alien](#)>

bullets [List](#) <[Bullet](#)>

score [Score](#)

# Class ObjectBase

Namespace: [SpicyInvaders](#)

Assembly: SpicyInvaders.dll

```
public class ObjectBase
```

## Inheritance

[object](#) ← ObjectBase

## Derived

[Alien](#), [Bullet](#), [Player](#)

## Inherited Members

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

# Properties

## \_color

```
public ConsoleColor _color { get; set; }
```

## Property Value

[ConsoleColor](#)

## \_isAlive

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

## Property Value

[bool](#)

## \_life



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

Property Value

[int](#)

**\_speed**

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

Property Value

[int](#)

**\_x**

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

Property Value

[int](#)

**\_y**

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

Property Value

[int](#)

## Methods

**\_IsAlive()**

Méthode qui renvoie si l'objet est toujours en vie en fonction du nombre de vies.

```
public bool _IsAlive()
```

Returns

[bool](#)

## \_OutLimitWindowYCheck(int)

Méthode qui renvoie si l'objet est en dehors de l'axe Y de la console.

```
public bool _OutLimitWindowYCheck(int windowHeight)
```

Parameters

**windowHeight** [int](#)

Returns

[bool](#)

# Class Player

Namespace: [SpicyInvaders](#)








Assembly: SpicyInvaders.dll

```
public class Player : ObjectBase
```

## Inheritance

[object](#)  ← [ObjectBase](#)  ← Player

## Inherited Members

[ObjectBase.x](#) , [ObjectBase.y](#) , [ObjectBase.color](#) , [ObjectBase.speed](#) , [ObjectBase.life](#) ,  
[ObjectBase.isAlive](#) , [ObjectBase.IsAlive\(\)](#) , [ObjectBase.OutLimitWindowYCheck\(int\)](#) ,  
[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,  
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Constructors

### Player(int, int, ConsoleColor, int, int)

Initialisation de l'objet et enregistrement de ses valeurs

```
public Player(int originX, int originY, ConsoleColor color, int speed, int life)
```

## Parameters

originX [int](#) 

originY [int](#) 

color [ConsoleColor](#) 

speed [int](#) 

life [int](#) 

## Fields

### canAttack

```
public bool canAttack
```

Field Value

[bool](#)

## dimensionX

```
public int dimensionX
```

Field Value

[int](#)

## dimensionY

```
public int dimensionY
```

Field Value

[int](#)

## moveAllowLeft

```
public bool moveAllowLeft
```

Field Value

[bool](#)

## moveAllowRight

```
public bool moveAllowRight
```

Field Value

[bool](#)

## nickName

```
public string nickName
```

## Field Value

[string](#)

## Methods

### Draw()

La méthode Draw() affichera la forme de le player avec different couleur en fonction de sa quantité de vie.

```
public void Draw()
```

### Move(bool, bool)

```
///
```

Méthode Move() exprimant le mouvement avec la vitesse sur l'axe des x et en fonction de le sens de mouvement

```
public void Move(bool moveLeft, bool moveRight)
```

## Parameters

```
moveLeft bool
```

```
moveRight bool
```

# Class Score

Namespace: [SpicyInvaders](#)

Assembly: SpicyInvaders.dll

Clas SCORE

```
public class Score
```

## Inheritance

[object](#) ← Score

## Inherited Members

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

# Constructors

## Score(int)

Méthode de score qui demande d'enregistrer le nombre initial de points

```
public Score(int score0)
```

## Parameters

score0 [int](#)

# Fields

## points

```
public int points
```

## Field Value

[int](#)

## pointsForDmg

```
public int pointsForDmg
```

Field Value

[int](#)

## pointsForEnemy

```
public int pointsForEnemy
```

Field Value

[int](#)

# Class Alien

Namespace: [SpicyInvaders](#)








Assembly: SpicyInvaders.dll

```
public class Alien : ObjectBase
```

## Inheritance

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

## Inherited Members

[ObjectBase.x](#) , [ObjectBase.y](#) , [ObjectBase.color](#) , [ObjectBase.speed](#) , [ObjectBase.life](#) ,  
[ObjectBase.isAlive](#) , [ObjectBase.IsAlive\(\)](#) , [ObjectBase.OutLimitWindowYCheck\(int\)](#) ,  
[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,  
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Constructors

### Alien(int, int, ConsoleColor, int, int)

```
public Alien(int originX, int originY, ConsoleColor colorSkin, int speed, int life)
```

## Parameters

originX [int](#) 

originY [int](#) 

colorSkin [ConsoleColor](#) 

speed [int](#) 

life [int](#) 

## Fields

### assetLimitX

```
public int assetLimitX
```



Field Value

[int](#)

## assetLimitY

```
public int assetLimitY
```

Field Value

[int](#)

## moveDown

```
public bool moveDown
```

Field Value

[bool](#)

## moveLeft

Variables spécifiques à Alien

```
public bool moveLeft
```

Field Value

[bool](#)

## Methods

### Draw()

La méthode Draw() affichera la forme de l'alien en fonction de sa quantité de vie.

```
public void Draw()
```

# Move()

Méthode Move() exprimant le mouvement avec la vitesse sur l'axe des x

```
public void Move()
```

# Class Bullet

Namespace: [SpicyInvaders](#)








Assembly: SpicyInvaders.dll

```
public class Bullet : ObjectBase
```

## Inheritance

[object](#)  ← [ObjectBase](#) ← Bullet

## Inherited Members

[ObjectBase.x](#), [ObjectBase.y](#), [ObjectBase.color](#), [ObjectBase.speed](#), [ObjectBase.life](#),  
[ObjectBase.isAlive](#), [ObjectBase.IsAlive\(\)](#), [ObjectBase.OutLimitWindowYCheck\(int\)](#),  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

## Constructors

Bullet(int, int, ConsoleColor, bool, int, int)

```
public Bullet(int originX, int originY, ConsoleColor bulletColor, bool player, int speed,  
int life)
```

## Parameters

originX [int](#) 

originY [int](#) 

bulletColor [ConsoleColor](#) 

player [bool](#) 

speed [int](#) 

life [int](#) 

## Fields

fromPlayer

## Variables spécifiques à Bullet

```
public bool fromPlayer
```

## Field Value

[bool](#) 

# Methods

## Draw()

La méthode Draw() affichera la forme de bullet et son couleur

```
public void Draw()
```

## Move()

Méthode Move() exprimant le mouvement avec la vitesse sur l'axe des y

```
public void Move()
```

# Class DataBaseConnect

Namespace: [SpicyInvaders](#)

Assembly: SpicyInvaders.dll

```
public class DataBaseConnect
```

## Inheritance

[object](#) ← DataBaseConnect

## Inherited Members

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

## Fields

### playersName

```
public List<string> playersName
```

#### Field Value

[List](#) <[string](#)>

### playersScore

```
public List<long> playersScore
```

#### Field Value

[List](#) <[long](#)>

## Methods

### Connection()

Méthode de test de la connexion

```
public void Connection()
```

## Highscore()

Méthode d'enregistrement des résultats du jeu avec le nom du joueur et le score.

```
public void Highscore()
```

## SavePoints(string, int)

Méthode d'enregistrement des points

```
public void SavePoints(string nickName, int points)
```

## Parameters

nickName [string](#)<sup>↗</sup>

points [int](#)<sup>↗</sup>

# Class ObjectBaseTests

Namespace: [SpicyInvaders.Tests](#)








Assembly: SpicyInvadersTests.dll

```
[TestClass]  
public class ObjectBaseTests
```

## Inheritance

[object](#)  ← ObjectBaseTests

## Inherited Members

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

## Methods

### \_IsAliveTest()

UnitTesting pour verifier si l'object il est mort

```
[TestMethod]  
public void _IsAliveTest()
```

### \_OutLimitWindowYCheckTest()

UnitTesting pour verifier si l'object est dans la console

```
[TestMethod]  
public void _OutLimitWindowYCheckTest()
```