Class GameApplication

java.lang.Object
    javafx.application.Application
        GameApplication

```
public class GameApplication
extends javafx.application.Application
```

THE SPACESHIT - Pure Awesome

GameApplication opens a game window of a Space Shooter kind of game.

BY: Piotr Kusnierz, Sebastian Jarsve, Inge Brochmann

Since:

2018

Nested Class Summary

Nested classes/interfaces inherited from class javafx.application.Application

`javafx.application.Application.Parameters`

Field Summary

Fields

Modifier and Type	Field and Description
`private java.util.ArrayList<javafx.scene.Node>`	`enemies`
`int`	`frameCounter`
`private boolean`	`paused`
`private javafx.scene.Node`	`player`
`private int`	`playerLives`
`private javafx.scene.layout.Pane`	`root`
`private int`	`sizeX`
`private int`	`sizeY`
`private javafx.animation.AnimationTimer`	`timer`

Fields inherited from class javafx.application.Application

STYLESHEET_CASPIAN, STYLESHEET_MODENA

Constructor Summary

Constructors

Constructor and Description
`GameApplication()`

Method Summary

All Methods **Static Methods** **Instance Methods** **Concrete Methods**

Modifier and Type	Method and Description
`private javafx.scene.Parent`	`createContent()` This method creates the root window, initializes the player object and creates the game loop.
`private void`	`ifColiding()` If the player collides with an enemy, the enemy should be removed and you will lose one of your lives.
`private javafx.scene.Node`	`initEnemy()` Creates an enemy object with a random size and location.
`private javafx.scene.Node`	`initPlayer()` Method for initializing the player object.
`static void`	`main(java.lang.String[] args)` The main method to run the Application.
`private void`	`resetPlayerPosition(javafx.scene.Node player)` Setting the player position to its initial location.
`private void`	`showMessage(java.lang.String msg)` Creates an animated message to the user.
`void`	`start(javafx.stage.Stage stage)` An overriden method of the Application.

private void	**update**() This methods runs every frame to update game state.

Methods inherited from class javafx.application.Application

getHostServices, getParameters, getUserAgentStylesheet, init, launch, launch, notifyPreloader, setUserAgentStylesheet, stop

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

timer

```
private javafx.animation.AnimationTimer timer
```

root

```
private javafx.scene.layout.Pane root
```

enemies

```
private java.util.ArrayList<javafx.scene.Node> enemies
```

player

```
private javafx.scene.Node player
```

playerLives

```
private int playerLives
```

paused

```
private boolean paused
```

sizeY

```
private final int sizeY
```

sizeX

```
private final int sizeX
```

frameCounter

```
public int frameCounter
```

Constructor Detail

GameApplication

```
public GameApplication()
```

Method Detail

createContent

```
private javafx.scene.Parent createContent()
```

This method creates the root window, initializes the player object and creates the game loop.

initPlayer

```
private javafx.scene.Node initPlayer()
```

Method for initializing the player object.

Returns:

Node

resetPlayerPosition

```
private void resetPlayerPosition(javafx.scene.Node player)
```

Setting the player position to its initial location.

Parameters:

`player: - Player instance of type Object`

initEnemy

```
private javafx.scene.Node initEnemy()
```

Creates an enemy object with a random size and location.

Returns:

`Node`

showMessage

```
private void showMessage(java.lang.String msg)
```

Creates an animated message to the user.

Parameters:

`msg: - Type of String, text to be shown.`

update

```
private void update()
```

This methods runs every frame to update game state.

ifColiding

```
private void ifColiding()
```

If the player collides with an enemy, the enemy should be removed and you will lose one of your lives.

start

```
public void start(javafx.stage.Stage stage)
           throws java.lang.Exception
```

An overriden method of the Application. This method runs once at application startup, and sets the stage and scene of the game, as well as adding a keyboard handler for input.

Specified by:

`start in class javafx.application.Application`

Parameters:

`stage: - A Stage object which acts as the main container.`

Throws:

`java.lang.Exception`

main

```
public static void main(java.lang.String[] args)
```

The main method to run the Application.