

Package [FunctionC_TomCatchJerry](#)

Class **GameMazeGUI**

[java.lang.Object](#)
[javafx.application.Application](#)
[FunctionC_TomCatchJerry.GameMazeGUI](#)

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

GameMazeGUI class sets up the interface and the background music for the game. The maze of initial state will be set up in SetGridPane method. When the game is started, each character (Tom and Jerry) is worked with its own thread. Multithreading ensures that Tom and Jerry can move simultaneously with different speeds. The updated movement of the character is shown on the screen by updateGridPane method. Interactions with the player are achieved by message display and the button.

Nested Class Summary

Nested classes/interfaces inherited from class [javafx.application.Application](#)

[javafx.application.Application.Parameters](#)

Field Summary

Fields

Modifier and Type	Field	Description
javafx.scene.paint.ImagePattern	block	
static final int	CELL_SIZE	
List < javafx.scene.shape.Rectangle >	cells	
int	entryIndex	
int	exitIndex	
javafx.scene.layout.GridPane	gridPane	
javafx.scene.paint.ImagePattern	JerryJerry	
javafx.scene.paint.ImagePattern	TomTom	

Fields inherited from class [javafx.application.Application](#)

[STYLESHEET_CASPIAN](#), [STYLESHEET_MODENA](#)

Constructor Summary

Constructors

Constructor	Description
GameMazeGUI()	

Method Summary

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

Modifier and Type	Method	Description
void	SetGridPane()	Set up the scene of the maze and store the spawn point of Jerry and Tom to their location respectively

```
void      start(javafx.stage.Stage primaryStage)

void      updatedGridPane(Character c,
                           javafx.scene.paint.ImagePattern imagePattern)
```

Update the grid pane after initialising the location of Tom and Jerry

Build up the GUI for the maze game.

Update the interface of the maze game to show the object movement Fill the grid where the character located with the corresponding image Repaint the last position of the character back to the original color of that grid If the lastPos is an BLOCK, do nothing Else if the lastPos is entry point, fill the grid with #F1CD85 Else if the lastPos is exit point, fill the grid with #808990 Else, fill the grid with white color which refers to a path

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 Details

CELL_SIZE

```
public static final int CELL_SIZE
```

See Also:
[Constant Field Values](#)

entryIndex

```
public int entryIndex
```

exitIndex

```
public int exitIndex
```

gridPane

```
public javafx.scene.layout.GridPane gridPane
```

cells

```
public List <javafx.scene.shape.Rectangle> cells
```

TomTom

```
public javafx.scene.paint.ImagePattern TomTom
```

JerryJerry

```
public javafx.scene.paint.ImagePattern JerryJerry
```

block

```
public javafx.scene.paint.ImagePattern block
```

Constructor Details

GameMazeGUI

```
public GameMazeGUI()
```

Method Details

SetGridPane

```
public void SetGridPane()
```

Set up the scene of the maze and store the spawn point of Jerry and Tom to their location respectively Update the grid pane after initialising the location of Tom and Jerry

updatedGridPane

```
public void updatedGridPane(Character c,  
                             javafx.scene.paint.ImagePattern imagePattern)
```

Update the interface of the maze game to show the object movement Fill the grid where the character located with the corresponding image Repaint the last position of the character back to the original color of that grid If the lastPos is an BLOCK, do nothing Else if the lastPos is entry point, fill the grid with #F1CD85 Else if the lastPos is exit point, fill the grid with #808990 Else, fill the grid with white color which refers to a path

Parameters:

c - The character moving on the maze

imagePattern - The image which represents the corresponding character on the maze

start

```
public void start(javafx.stage.Stage primaryStage)
```

Build up the GUI for the maze game. Multithreading allows different events can work on different characters concurrently. The game interface has to update once the character's movement is finished. The thread should keep track on the game state. When the game is ended, no further actions on the corresponding characters are allowed. Home button always exist on the GUI. The game is forced to terminate if the game does not ended yet. The restart button and the end game message will only appear when the game ends. Reset the status of all characters and the game interface as initial state.

Specified by:

start in class javafx.application.Application

Parameters:

primaryStage - The stage shown on the screen