POO - Lab 11 - Properties

Instituto Politécnico de Setúbal - Escola Superior de Tecnologias de Setúbal

Object Oriented Programming

Degree in Computer Engineering 2022/2023

Tools

- BlueJ
- JavaFX

Objectives

- Introduction to using JavaFX

Programs

- Properties

Implementation Rules

- Create the application using the BlueJ IDE.
- Implementation of the required code and testing at the end of each level.
- Use coding conventions adopted for the Java language (see **Notes**).

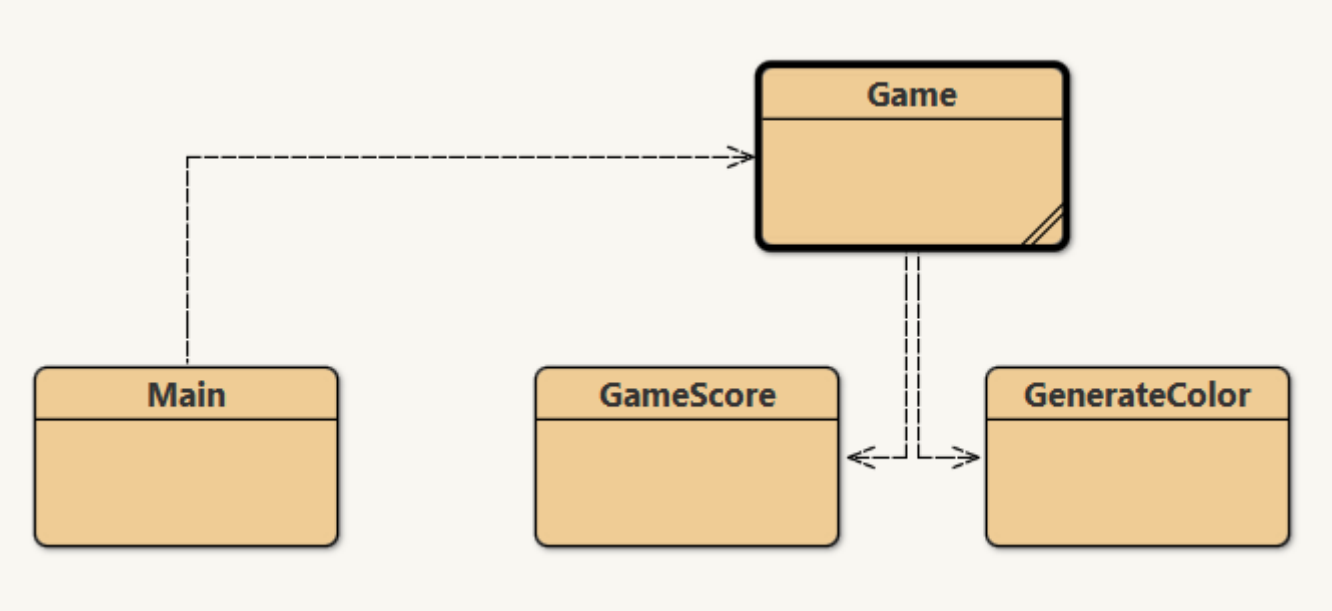
Introduction

GitHub Classrooms link: https://classroom.github.com/a/1jAo7czY

The goal of this lab will be to deepen your knowledge of JavaFX, using Object properties coming from the JavaFX library.

Download the assignment from lab 11 to a board on your computer and open the BlueJ project.

You can analyze the following the classes in BlueJ and at the end of the lab you will have the structure present in figure 1.



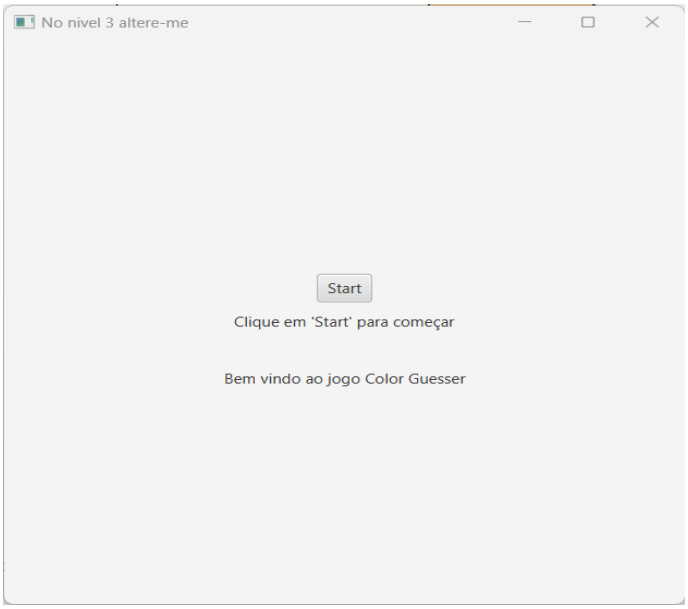
1. Open the contents of the GenerateColor and GameScore classes and comment on their methods in order to understand what each of these auxiliary classes do.
2. Do the same process for the Game class.

Level 1

The Main class is where the code for the JavaFX application is located. As you can see, not all of the code is present in the start method. You can see that there is a function in addition to the start method, it is called getInitialScene() and its function is to return the initial scene.

1. Let's create a Stage attribute in the Main class and now in the start method we will associate the Stage received in the start function with the Stage we defined earlier.
2. Next create a void function named changeScene that receives an instance of class Scene. This function will invoke the changeScene function from the stage attribute, and receive the scene that was given at the beginning of the function.
3. In the start method invoke the function created earlier with the Scene returned by the getInitialScene() function;

Expected result:



Level 2

1. Change the Scene title in the getInitialScene() function to "Welcome to the Color Guesser".
2. In Figure 2 you saw that the application works but doesn't look appealing. Change the Scene so that the label "Welcome" appears first, followed by "Click Start" and finally the button.
3. Next add the StackPane Id "sp" to "background" by using the following code:

```
sp.setId("background");

scene.getStylesheets().addAll(this.getClass().getResource("styles/style.css").toExternalForm());
```

This will assign the id "background" to the stackPane sp. Also add the styles contained in the styles/style.css file that will assign the gif to the background element. To make the letters show through better, change their color to white and change the Welcome label to the "Arial" font size 30.

Expected result

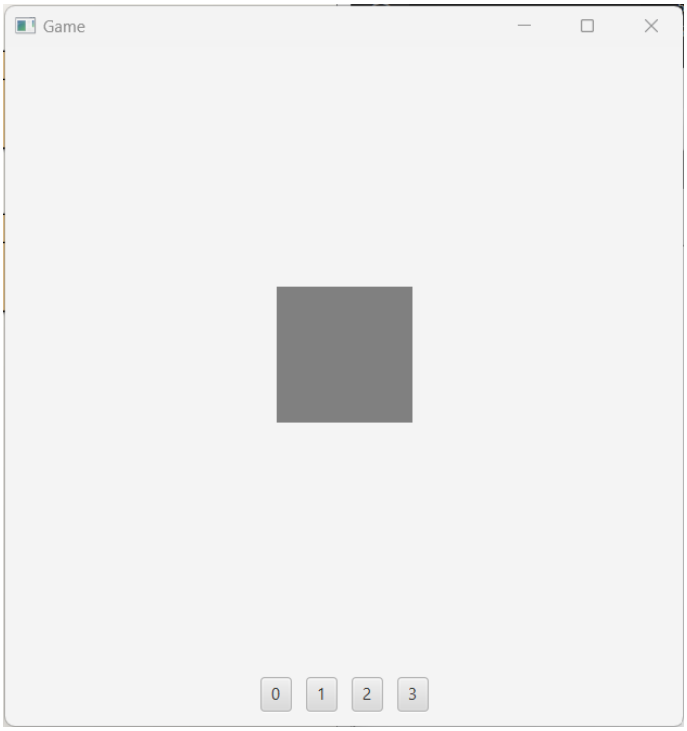
Level 3

Now with the first Scene created, we have an introduction to the Game. But there must be some way to switch to the Game Scene.

1. To do this, create a new function called getGameScene that returns a class of type Scene, similar to the getInitialScene method.
2. Start by creating in the getGameScene method, a BorderPane that will serve as the base for the new Scene. use the same dimensions as in the getInitialScene method of 500 pixels by 500 pixels minimum size of the BorderPane through the following method:

```
setMinSize(500, 500);
```

3. Create a HBox with four buttons. Each button should have the text 0, 1, 2, 3 as shown in Figure 3 and add it to the BorderPane at the "Bottom" position.
4. Add a rectangle with the Color "Gray" in the center of the BorderPane.



5. Finally add in the function getInicialScene, inside the EventHandler, btnStart handle the function changeScene with the Scene returned by the function getGameScene.

6. Change the title of the Scene to "Game".

Level 4

1. Add a property "game" from the Game class to the Main class, and instantiate it in the Start method. Now with the game attribute created use it to collect the game result through the getScore() method and put it in a label in the top left corner of the BorderPane. At first it will have a value of "0/0".

2. Next, change the color of the Rectangle from gray to the color returned by the getCorrectColor method of the game attribute.

Level 5

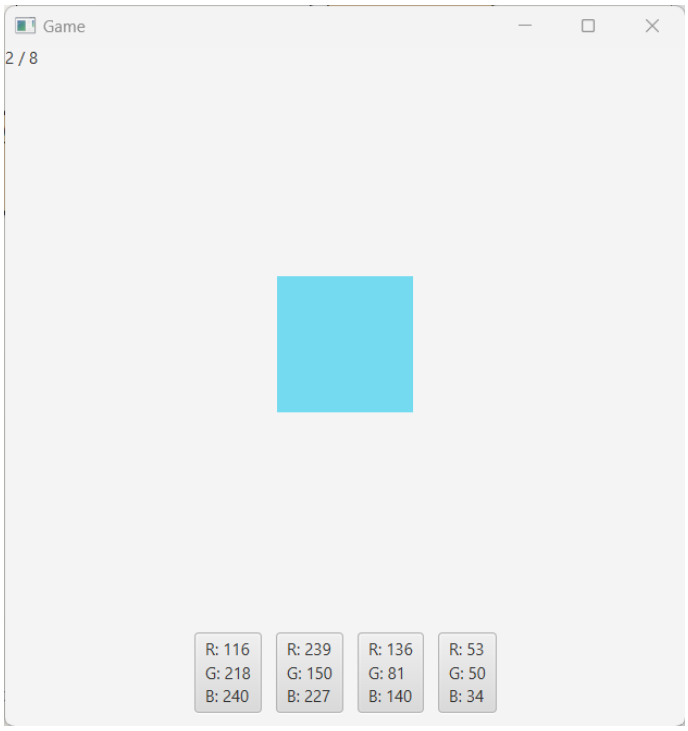
1. Next create an ArrayList of buttons that will now hold the four buttons instead of the ones you created earlier. Also create an ArrayList of colors, which will be the array returned by the getColors method of the game attribute.

2. Make a for loop that for each iteration picks up the color at position i from the color array, creates a button with the rgb values of that color (Warning! Multiply the values by 255) and finally adds the new button to the ArrayList of buttons.

3. Use the setOnAction method of each button so that when they are clicked on, they invoke the guess method of the game attribute with the respective color from the ArrayList of colors as input of the guess method, invoke the generateColors method of the same attribute and finally, change the Scene to the Scene returned by the getGameScene() method.

4. Finally, change the HBox to add all the buttons in the arrayList

Expected result:



Notes:

For identifiers follow commonly adopted conventions, in particular:

1. The camelCase notation for local variable names and identifiers for attributes and identifiers.
2. The PascalCase notation for class names.
3. Do not use the '_' symbol or abbreviations for identifiers.