

# POO - Lab 11 - Propriedades

---

Instituto Politécnico de Setúbal - Escola Superior de Tecnologias de Setúbal

Programação Orientada por Objetos

Licenciatura Engenharia Informática 2022/2023

## Ferramentas

- BlueJ
- JavaFX

## Objetivos

---

- Introdução ao uso de JavaFX

## Programa

- Propriedades

## Regras de Implementação

- Criar a aplicação utilizando o IDE BlueJ.
- Implementar o código necessário e testar no fim de cada nível.
- Use as convenções de codificação adotadas para a linguagem Java (ver **Notas**).

## Introdução

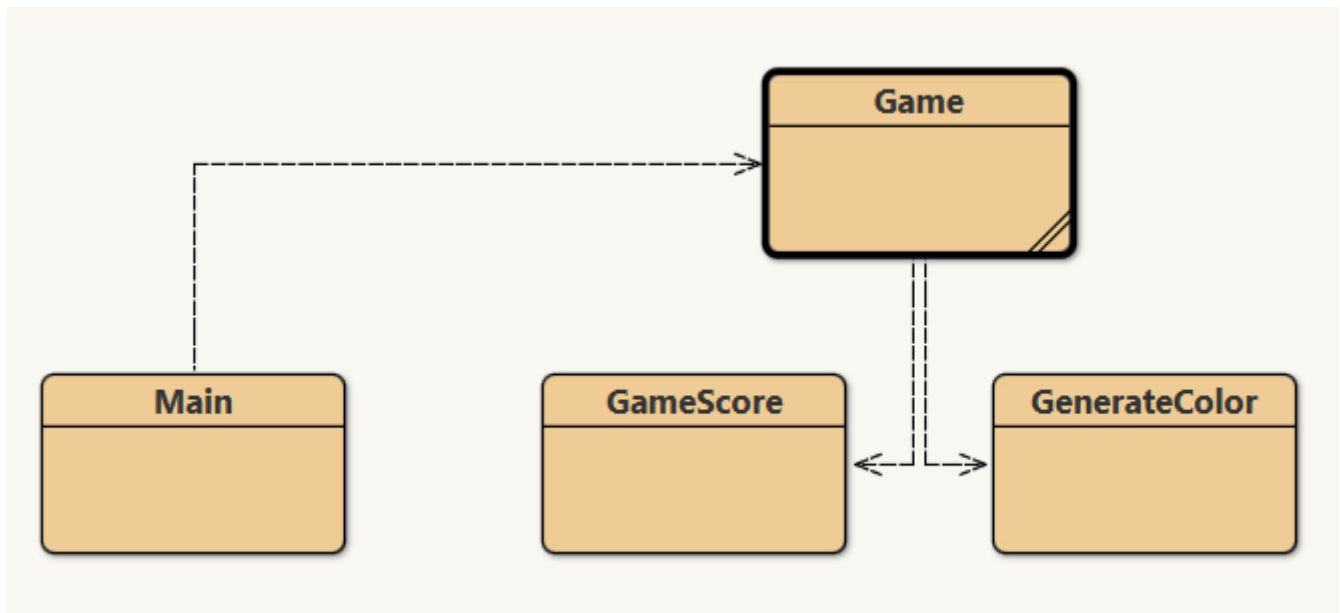
---

Link para GitHub Classrooms: <https://classroom.github.com/a/1jAo7czY>

O objetivo deste laboratório será de aprofundar os conhecimentos de JavaFX, utilizando propriedades de Objetos provenientes da biblioteca JavaFX.

Faça o colonamento do assignment do laboratório 11 para uma diretoria do seu computador e abra o projeto BlueJ.

É possível analisar a seguinte as classes no BlueJ e no final do laboratório terá a estrutura presente na figura 1.



1. Abra o conteúdo das classes GenerateColor e GameScore e comente os métodos das mesmas de forma a perceber o que cada uma destas classes auxiliares fazem.
2. Faça o mesmo processo para a classe Game.

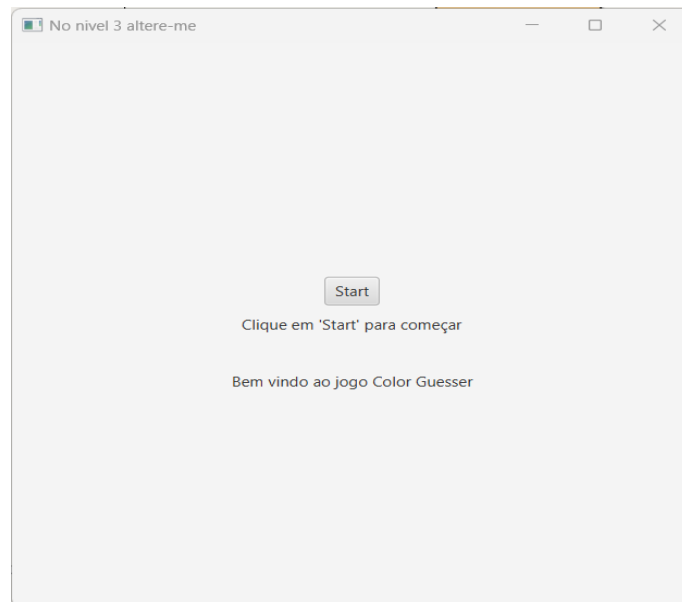
## Nível 1

---

Na classe Main é onde se encontra o código da aplicação JavaFX. Como pode ver, nem todo o código está presente na método start. Pode-se ver que existe uma função para além do método start, esta chama-se `getInicialScene()` e tem como função retornar a cena inicial.

1. Vamos criar um atributo Stage na class Main e agora no método start vamos associar a Stage recebida na função start á Stage que definimos anteriormente.
2. De seguida crie uma função sem retorno de valores (void) com o nome de `changeScene` e que recebe uma instância da class Scene. Esta função invocará a função `changeScene` do atributo stage, e receberá a scene que foi dada no inicio da função.
3. No método start invoque a função criada anteriormente com a Scene retornada pela função `getInicialScene()`;

Resultado esperado:



## Nível 2

1. Altere o título da Scene na função `getInicialScene()` para "Welcome to the Color Guesser".
2. Na figura 2 viu-se que a aplicação funciona mas Não está com um visual apelativo. Altere a Scene de modo a que a label "Bem Vindo" apareça primeiro, seguida da "clique em Start" e finalmente o botão.
3. De seguida adicione o Id da StackPane "sp" para "background" através do seguinte código:

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

Isto irá atribuir o id "background" á stackPane sp. Adicione também os estilos contidos no ficheiro `styles/style.css` que irá atribuir o gif ao elemento de background. Para que as letras se vejam melhor, altere a cor das mesmas para branco e altere a label Welcome para o tipo de letra "Arial" tamanho 30.

## Resultado esperado



## Nível 3

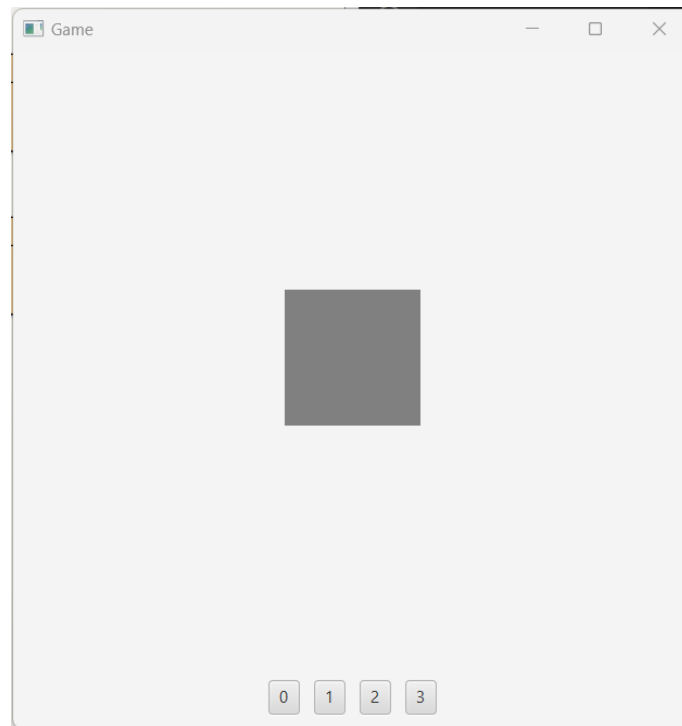
---

Agora com a primeira Scene criada, temos uma introdução ao Jogo. Mas é necessário que haja alguma maneira de mudar para a Scene do jogo.

1. Para isso crie uma nova função com o nome de `getGameScene` que retorna uma classe do tipo `Scene`, de forma similar ao método `getInicialScene`.
2. Comece por criar no método `getGameScene`, uma `BorderPane` que servirá de base para a nova `Scene`. utilize as mesmas dimensões tal como no método `getInicialScene` de 500 pixels por 500 pixels de tamanho mínimo da `BorderPane` através do seguinte método:

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

3. Crie uma `HBox` com quatro botões. Cada botão deverá ter o texto 0, 1, 2, 3 como mostra a figura 3 e adicione-a à `BorderPane` na posição "Bottom".
4. Adicione um rectângulo com a Cor "Cinza" no centro da `BorderPane`.



5. Finalmente adicione na função `getInitialScene`, dentro do `EventHandler`, handle do `btnStart` a função `changeScene` com a `Scene` retornada pela função `getGameScene`.
6. Altere o título da `Scene` para "Game"

## Nível 4

---

1. Adicione uma propriedade "game" da Classe `Game` na classe `Main`, e instancie a mesma no método `Start`. Agora com o atributo `game` criado utilize o mesmo para recolher o resultado do jogo através do método `getScore()` e coloque o mesmo numa label no topo superior esquerdo da `BorderPane`. No início o mesmo terá o valor de "0/0".
2. De seguida, altere a cor do Rectângulo de cinzento para a Cor retornada pelo método `getCorrectColor` do atributo `game`.

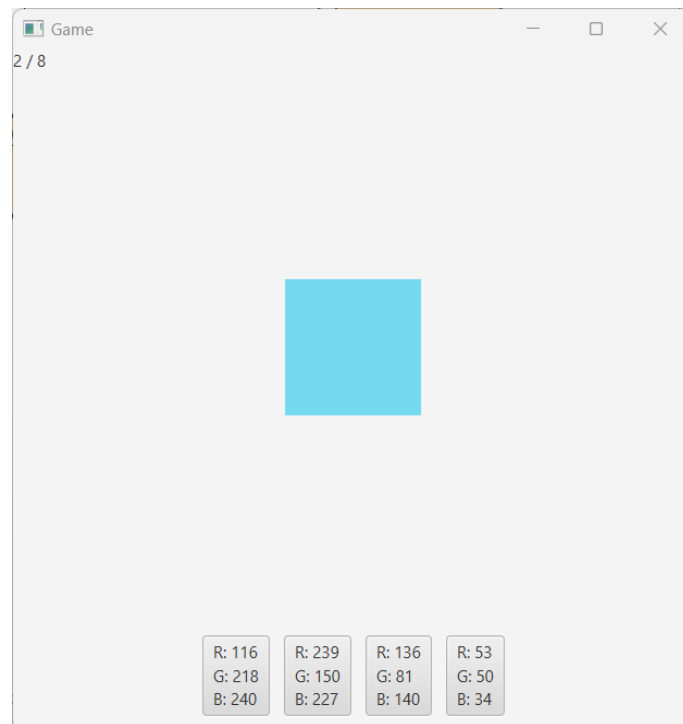
## Nível 5

---

1. De seguida crie um `ArrayList` de botões que irá guardar agora os quatro botões ao invés dos que foram criados anteriormente. Crie também um `ArrayList` de cores, que será o array devolvido pelo método `getColors` do atributo `game`.
2. Faça um ciclo `for` que para cada iteração recolhe a cor na posição `i` do array de cores, crie um botão com os valores `rgb` dessa cor (Aviso! Multiplique os valores por 255) e por fim adicione o novo botão ao `ArrayList` de botões.
3. Utilize o método `setOnAction` de cada botão para que quando sejam clicados, invoquem, o método `guess` do atributo `game` com a respetiva cor presente no `ArrayList` de cores como input do método `guess`, invoque o método `generateColors` do mesmo atributo e finalmente, altere a `Scene` para a `Scene` retornada pelo método `getGameScene()`.

4. Finalmente, altere a HBox para adicionar todos os botões no arrayList

Resultado esperado:



Notas:

Para os identificadores siga as convenções adotadas normalmente, em particular:

1. A notação camelCase para o nome das variáveis locais e identificadores de atributos e métodos.
2. A notação PascalCase para os nomes das classes.
3. Não utilize o símbolo '\_', nem abreviaturas nos identificadores.