

Fundamentos de Java

3-8: Animação do Mundo e Final do Jogo

Projeto:

Objetivos da Lição – Usar imagens e animação

- Adicionar uma nova propriedade/campo
- Modificar um método
- Detectar uma colisão

Instruções:

Abra o cenário da lição 7 (JF_V02_S03_L07PrjStudent).

Execute as seguintes tarefas:

1. Salve o cenário como JF_V02_S03_L08PrjStudent
2. Na classe de robô adicione dois campos
 - a. `private GreenfootImage robotimage1;`
 - b. `private GreenfootImage robotimage2;`
3. Crie um método construtor para a classe Robot que atribua as duas imagens robot da seguinte forma:
 - a. `robotimage1= new GreenfootImage("man01.png");`
 - b. `robotimage2= new GreenfootImage("man02.png");`
4. No Robô, crie um novo método public void denominado `animate()`.
5. Em `animate()` crie um código que represente o pseudocódigo a seguir.

```
If current image displayed equals robot1 image then
    set image as robot2
Else
    set image as robot1
```
6. Adicione chamadas para animar dentro do `robotMovement` de modo que o robô se mova e chame `animate`. O robô só deve animar ao mover-se.
7. Crie uma propriedade no Robô para armazenar o número de vidas. Defina o valor das vidas como 3 no construtor.
8. Crie uma propriedade no Robô para armazenar a pontuação atual. Defina a pontuação como 0 no construtor.
9. Crie uma propriedade no Robô denominada `pizzaEaten` para armazenar o número de pizzas comidas. Defina `pizzaEaten` como 0 no construtor.
10. Modifique o método `eatPizza` no Robô de modo que o número de `pizzaEaten` seja incrementado por 1 para cada pizza comida.

11. Modifique `detectHome` para só reproduzirmos o arquivo e redefinirmos a localização do robô se todas as instâncias da Pizza tiverem sido comidas quando colidirmos com a casa. Também redefina o contador de pizza para 0.
12. No Robô, crie um método denominado `removeLife()`. Codifique isso de modo que o número de vidas seja reduzido em um.
13. Adicione a chamada do método `removeLife` a `detectBlockCollision` e `detectWallCollision` de modo que, se eles colidirem, uma vida seja removida.
14. Adicione um método denominado `testEndGame()` ao Robô. Codifique-o de modo que, se o número de vidas for menor ou igual a 0, o jogo termine. Adicione esse método a `removeLife`.
15. Adicione outra propriedade ao Robô que armazene uma imagem denominada `gameover.png`
16. Modifique `testEndGame()` de modo que antes que o jogo termine a imagem do robô seja alterada para `gameoverimage`
17. Crie um método denominado `increaseScore` que acrescente 1 ponto à pontuação. Adicione essa chamada de método à parte inferior do método `detectHome`.
18. Crie um método denominado `showStatus` que exiba o número de vidas e a pontuação no painel de pontuação.
19. Modifique `removeLife` e `increaseScore` para chamar `showStatus`.

Advanced Task – Tente desacelerar a animação do robô usando um contador para só alterar a imagem depois que 5 atos forem executados.

20. Compile seu cenário.
21. Salve o cenário.