

Um simulador 2D para a linguagem Robotoy

Diogo da Silva

Orientadora: Joyce Martins

Roteiro

- Introdução;
- Objetivos;
- Fundamentação:
 - Linguagens interpretadas e *scripts*;
 - Simulação;
 - Trabalhos Correlatos.
- Requisitos não funcionais;
- Especificação (*script* e interface);
- Implementação;
- Resultados, discussões, conclusões e extensões.



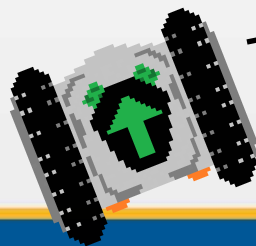
Introdução

- Robótica é uma das apostas das universidades;
- Desenvolvimento do raciocínio lógico e da capacidade de resolver problemas;
- Robotoy (TORRENS, 2014) surge a fim de facilitar o desenvolvimento em robôs Lego;
- Simulação x acessibilidade.



Objetivos

- **Geral:** desenvolver uma ferramenta para simular as ações de um robô em cenários 2D utilizando a linguagem Robotoy.
- Os objetivos **específicos** do trabalho são:
 - disponibilizar uma interface para criação e edição dos cenários 2D;
 - disponibilizar uma interface para a edição de programas na linguagem Robotoy;
 - executar os programas Robotoy nos cenários previamente criados.

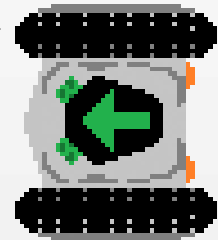


Pera aí, quer dizer que eu não fui planejado?

Fundamentação Teórica

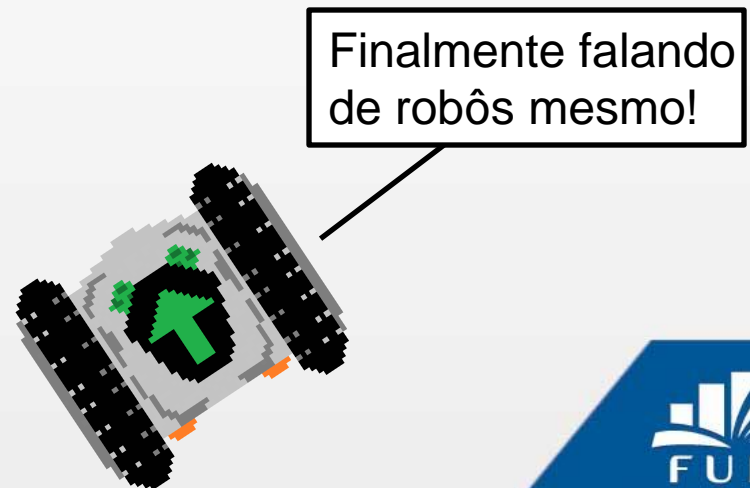
- Linguagens interpretadas e interpretadores;
- Cada interpretador possui suas próprias rotinas internas;
- *Scripts*:
 - sequência de comandos;
 - interpretador (*browser*);
 - não produzem código intermediário;
 - compilador x interpretador.

Eu sou quase um interpretador então?

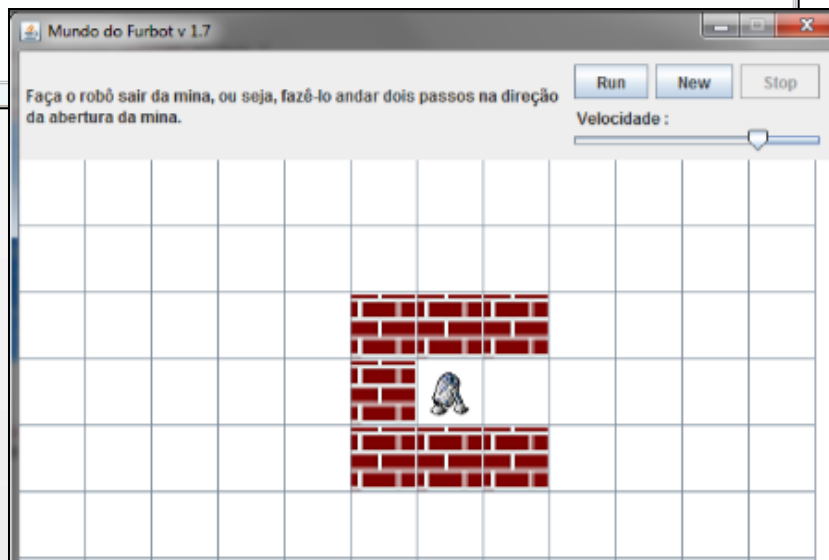


Fundamentação Teórica

- Simulação;
- Processo de modelar um sistema real;
- Robótica e simulação atuam juntas;
- Na elaboração de um simulador para robôs, pode-se definir cinco passos:
 - gatilho;
 - lógica;
 - timer;
 - máquina de execução;
 - alvo.

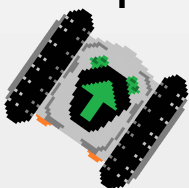


Trabalhos Correlatos




```
while ( true ) {  
  cannon( 315 , 40 );  
  if ( scan( 315 ) > 39 ) {  
    swim( 315 );  
  } else {  
    stop();  
  }  
}
```

Família!



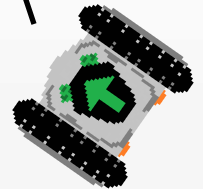
Trabalhos Correlatos

características	Furbot	Robomind FURB	Blockly	Simulador Robotoy
possui linguagem de alto nível simplificada e de fácil assimilação por usuários não programadores		X	X	X
possui linguagem intermediária acessível e que pode ser gerada a partir de qualquer linguagem de alto nível			X	X
possui ambiente de programação próprio		X	X	X
tipo de programação	textual	textual	visual	textual
permite simular os comandos em robôs virtuais	X	X	<div data-bbox="1460 1039 1599 1115" style="border: 1px solid black; padding: 2px;">Sim.</div> 	
permite executar os comandos em robôs reais		X		
possui interface para editar cenários de simulação		X		X

Requisitos não funcionais

- ser implementado na linguagem Java;
- ser desenvolvido na IDE Eclipse;
- utilizar a biblioteca OpenGL para Java (JOGL) para modelar os cenários 2D;
- utilizar a linguagem Robotoy como linguagem de alto nível para programar as ações dos robôs que serão simuladas.

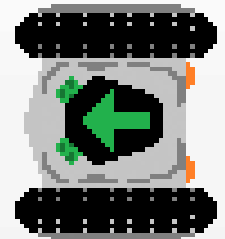
Possuir um robô lindo que executa os comandos.
Não?



Linguagem intermediária

- Desenvolvida uma linguagem intermediária, um *script*;
- Facilitar a adaptação a outras linguagens;
- Comandos divididos em:
 - executados no robô;
 - controle de variáveis;
 - controle de fluxo e rotinas.
- Interpretador desse *script* é quem controla a simulação.

Robô que no caso sou eu, o Billy.

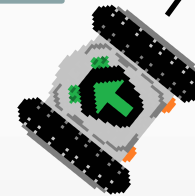


Linguagem intermediária

- Exemplos de comandos:
 - Movimentar o robô para frente

```
COMANDO_ROBO;EM_FRENTE;4
```

Entenderam?
Eu entendo!



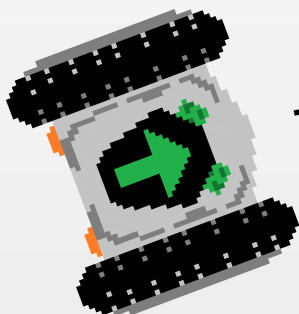
- Identificar cores

```
DECLARAR_VARIAVEL;cor_identificada;'BRANCO'  
[...]  
COR_IDENTIFICADA  
DECLARAR_VARIAVEL;c;cor_identificada
```

Linguagem intermediária

- Exemplos de comandos:
 - Comando de repetição (*while*)

```
DECLARAR_VARIAVEL;n;1  
DECLARAR_VARIAVEL;t;""  
WHILE;n <= 10  
ATRIBUIR_VALOR_VARIAVEL;t;"3 x" + n + "=" + n * 3  
ESCREVER;( t )  
ATRIBUIR_VALOR_VARIAVEL;n;n + 1  
ENDWHILE
```

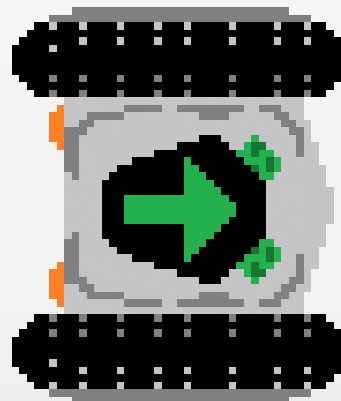


3 x 1 = 3

...

Simulador

- Dividido em três partes:
 - Compilador;
 - Editor de cenários;
 - Controle de simulação.



Aí vamos nós!

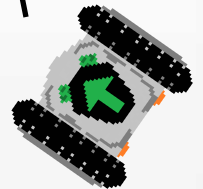
Simulador

8 escrever bt
9
10 rotina addValor
11 valor <- valor + 10
12 virar motor multiuso para a direita 1
13 se não tem obstáculo
14 andar para frente
15 virar motor multiuso para a esquerda 1
16 fim do se
17 fim da rotina
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39. At the bottom left, it says 'Valor: 60' and 'Execução finalizada: Fim do arquivo alcançado.'"/>

Resultados e Discussões

- Para editar os cenários é preciso apenas uma instrução básica (funções sem botão);
- Não é recomendável utilizar cenários maiores que 35x35;
- Atende ao requisito de apresentar um facilitador para a utilização da Robotoy.

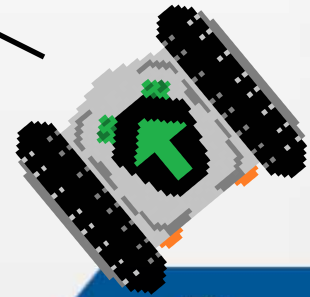
35x35? Eu teria que andar bastante...



Conclusões

- Cumpre os objetivos por:
 - oferecer interfaces para criar e editar cenários 2D, assim como programas Robotoy;
 - possibilitar simular os programas Robotoy compilados.
- A linguagem intermediária (*script*) foi de suma importância;
- Problemas com o JOGL em relação ao carregamento de texturas.

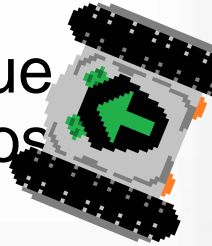
Tem um robô
lindo...



Extensões

3D? Eu? Que loucura cara!

- a) substituir o editor de cenários 2D por um 3D;
- b) adicionar os comandos da linguagem Robotoy que foram suprimidos, além de incluir comandos novos como pegar, presente no Robomind FURB;
- c) adicionar outras linguagens ao compilador, para que os usuários possam optar por diferentes linguagens fonte;
- d) melhorar o desempenho do simulador em relação a execução em cenários muito grandes;
- e) criar uma ferramenta complementar com a finalidade de tornar possível que os usuários criem suas próprias texturas e adicionem-as ao simulador.



O Billy agradece!

Fácil né galera?

