



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO



S-Educ: Um Simulador de Ambiente de Robótica Educacional em Plataforma Virtual

Carla da Costa Fernandes

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Natal, RN, 30 de janeiro de 2013

UFRN / Biblioteca Central Zila Mamede
Catalogação da publicação na fonte

Fernandes, Carla da Costa.

S-Educ: Um Simulador de Ambiente de Robótica Educacional em Plataforma Virtual / Carla da Costa Fernandes - Natal, RN, 2013

89 f. : il

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves

Dissertação (Mestrado) - Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica e de Computação.

1. Robótica educacional - Dissertação.
 2. Simulador - Robótica - Dissertação.
 3. Ambiente virtual - Dissertação.
 4. Engenharia Elétrica e da Computação - Dissertação.
- I. Gonçalves, Luiz Marcos Garcia. II. Universidade Federal do Rio Grande do Norte. III. Título.

RN/UF/BCZM
CDU 004.896:37

S-Educ: Um Simulador de Ambiente de Robótica Educacional em Plataforma Virtual

Carla da Costa Fernandes

Dissertação de Mestrado aprovada em 30 de janeiro de 2013 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Luiz Marcos Garcia Gonçalves (orientador) DCA/UFRN

Prof. Dr. Eduardo Oliveira Freire DEL/UFS

Prof. Dr. Aquiles Medeiros Filgueira Burlamaqui ECT/UFRN

Agradecimentos

Ao meu orientador, professor Luiz Marcos Garcia Gonçalves, sou grata pela orientação e confiança depositados em minha pessoa.

Aos meus pais, Flávio e Silvia, e à minha irmã Flávia, pelo apoio durante esta jornada.

Ao meu namorado, Danilo, pelas sugestões e críticas no desenvolvimento deste trabalho.

À equipe RoboEduc, pelas contribuições dadas neste trabalho.

A toda a equipe do laboratório NatalNet.

Aos Top&Amigos.

Ao CNPq, pelo apoio financeiro.

Resumo

Propomos uma plataforma de simulação robótica, denominada S-Educ, desenvolvida especificamente para aplicação em robótica educacional, que pode ser usada como uma alternativa ou em conjunto com kits de robôs em aulas envolvendo uso da robótica. Na abordagem usualmente conhecida, a robótica educacional utiliza-se de *kits* de robótica em aulas que geralmente contemplam temas interdisciplinares. A ideia deste trabalho não é substituir esses kits, mas sim usar o simulador desenvolvido como uma alternativa, onde, por algum motivo, os kits tradicionais não possam ser usados, ou até mesmo usar a plataforma em conjunto com esses kits. Para o desenvolvimento dos trabalhos, realizamos inicialmente uma pesquisa bibliográfica acerca da utilização de simuladores robóticos voltados para a área educacional, e de kits robóticos, a partir da qual foi possível definir um conjunto de funcionalidades consideradas importantes para a criação de tal ferramenta. Em seguida, na fase de desenvolvimento de *software*, foi implementado o simulador S-Educ, levando em consideração os requisitos e funcionalidades definidos na fase de projeto. Por fim, para validar a plataforma, foram realizados vários testes com professores, alunos e adultos leigos, nos quais foi utilizado o simulador S-Educ, visando avaliar a sua utilização em aulas de robótica educacional. Os resultados mostram que simuladores robóticos possibilitam uma redução de custos financeiros, facilitam os testes e diminuem os danos, devido ao uso, no robô, além de outras vantagens. Além disso, como contribuição à comunidade, a ferramenta proposta pode ser usada para aumentar a adesão das escolas brasileiras às metodologias de robótica educacional, ou em competições de robótica.

Palavras-chave: Simulador, Robótica Educacional, Ambiente virtual.

Abstract

We propose a robotics simulation platform, named S-Educ, developed specifically for application in educational robotics, which can be used as an alternative or in association with robotics kits in classes involving the use of robotics. In the usually known approach, educational robotics uses robotics kits for classes which generally include interdisciplinary themes. The idea of this work is not to replace these kits, but to use the developed simulator as an alternative, where, for some reason, the traditional kits cannot be used, or even to use the platform in association with these kits. To develop the simulator, initially, we conducted research in the literature on the use of robotic simulators and robotic kits, facing the education sector, from which it was possible to define a set of features considered important for creating such a tool. Then, on the software development phase, the simulator S-Educ was implemented, taking into account the requirements and features defined in the design phase. Finally, to validate the platform, several tests were conducted with teachers, students and lay adults, in which it was used the simulator S-Educ, to evaluate its use in educational robotics classes. The results show that robotic simulator allows a reduction of financial costs, facilitate testing and reduce robot damage inherent to its use, in addition to other advantages. Furthermore, as a contribution to the community, the proposed tool can be used to increase adhesion of Brazilian schools to the methodologies of educational robotics or to robotics competitions.

Keywords: Simulator, Educational Robotics, Virtual Environment.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
1 Introdução	1
1.1 Escopo do Trabalho	2
1.2 Motivação	3
1.3 Metodologia	5
1.4 Aplicações	5
1.5 Estrutura do trabalho	7
2 Fundamentação Teórica	9
2.1 A Robótica Educacional	9
2.2 Ferramentas utilizadas em aulas de robótica educacional	11
2.2.1 <i>Hardware</i>	12
2.2.2 <i>Software</i>	13
2.3 Simuladores Robóticos	14
2.3.1 Simuladores Robóticos na Educação	16
2.4 Tecnologias Utilizadas	17
3 Trabalhos Relacionados	19
3.1 Simuladores para Robótica Educacional	19
3.1.1 Simuladores para montagem de robôs	19
3.1.2 Simuladores para programação de robôs	20
3.1.3 Simuladores para montagem e programação de robôs	22
3.2 Análise Comparativa	24
4 Desenvolvendo o Simulador S-Educ	29
4.1 <i>Software</i> Educacional R-Educ	30

4.2	Modelagem do robô virtual	30
4.2.1	Movimentos do robô	30
4.2.2	Sensores	32
4.2.3	Atuadores	34
4.2.4	Detecção de Colisões	35
4.3	Interface Gráfica	36
4.4	Diagramas <i>UML</i>	37
4.4.1	Diagrama de Classes	37
4.4.2	Diagrama de Casos de Uso	41
4.4.3	Diagrama de Atividades	46
5	Experimentos e Resultados	49
5.1	Plataforma de testes	49
5.2	Testes iniciais	50
5.3	Grupos de testes	51
5.3.1	Experimentos com professores	52
5.3.2	Experimentos com alunos	54
5.3.3	Experimentos com leigos	56
5.4	Análise dos Resultados	57
6	Conclusão	59
Referências bibliográficas		62
A Questionário 1: Professores		69
B Questionário 2: Alunos		71
C Questionário 3: Leigos		73

Lista de Figuras

2.1	Exemplo de uma aula de robótica educacional.	10
3.1	Simuladores para programação de robôs.	22
3.2	Simuladores para montagem e programação de robôs.	25
4.1	Bases modeladas para o simulador S-Educ.	31
4.2	Sensores virtuais.	34
4.3	Sensores dos robôs virtuais.	34
4.4	Tipos de atuadores do simulador S-Educ.	35
4.5	Modelo de um robô com <i>bounding volume</i>	36
4.6	Tela de escolha entre aluno e professor	38
4.7	Telas de atividades do simulador S-Educ.	39
4.8	Telas do ambiente virtual do simulador S-Educ.	40
4.9	Diagrama de Classes do Simulador S-Educ.	41
4.10	Diagrama de Casos de Uso do Simulador S-Educ.	42
4.11	Visualização 3D e 2D do mesmo ambiente virtual.	44
4.12	Possíveis localizações do sensor ultrasônico na montagem do robô virtual.	45
4.13	Diagrama de atividades do simulador S-Educ.	47
5.1	Atividade 1: Labirinto.	51
5.2	Atividade 2: Múltiplos Robôs.	52
5.3	Atividade 3: Mapa.	52
5.4	Atividade 4: Seguir Linha.	53
5.5	Atividade 5: Células-Tronco.	53
5.6	Gráficos das pesquisas feitas com o grupo de professores.	55
5.7	Gráficos das pesquisas feitas com o grupo de alunos.	56
5.8	Gráfico da pesquisa feita com o grupo de adultos leigos em robótica.	57

Lista de Tabelas

3.1	Tabela comparativa entre os Simuladores de Robótica Educacional.	26
4.1	Exemplos de comandos em linguagem R-Educ.	30
4.2	Velocidades dos diferentes tipos de bases.	31
5.1	Tempo para a realização da atividade 3.	57

Capítulo 1

Introdução

No atual modelo de convivência social, a tecnologia deve estar sempre acessível e deve permitir às pessoas novas experiências, novas descobertas e novas formas de aprender. Contudo, a fim de que a população e, em particular, estudantes possam usufruir das ferramentas tecnológicas existentes, é necessário que o processo educativo inclua práticas tecnológicas em seu contexto educacional. Na busca por ambientes de aprendizagem mais ricos e inovadores, a robótica educacional se destaca por mostrar, na prática, conceitos teóricos e por desenvolver competências como raciocínio lógico, investigação e resolução de problemas [Miranda et al. 2010]. Segundo Prado (2008), as escolas devem iniciar o processo de alfabetização tecnológica, no qual o uso de tecnologias como a robótica são utilizados para transformar a vida escolar em um ambiente mais desafiador, criativo e dinâmico, que viabiliza a construção de um conhecimento crescente baseado em experimentações. A robótica educacional incentiva criação e exploração de ambientes interativos para o processo de ensino e aprendizagem no estudo das diversas disciplinas.

Este ambiente de aprendizagem une a robótica e a educação, através da multidisciplinaridade da robótica, que é capaz de envolver temáticas relacionadas a ela, como mecânica, eletrônica e computação, e também temáticas não diretamente relacionadas a ela, como matemática, ciências, línguas, ciências sociais, entre outras. De acordo com Zilli (2004), a robótica educacional ensina o aluno através da investigação, estimulando a criatividade, no desenvolvimento de um método científico, através do erro construtivo, estímulo da lógica, aprimoramento da motricidade, entre outros. Segundo Silva (2009), uma das metodologias adequadas para o desenvolvimento de aulas de robótica educacional em sala de aula é através de oficinas de robótica, nas quais os alunos seguem um conjunto de passos: análise do desafio, montagem do protótipo robótico e programação do robô. O desafio proposto pode envolver assuntos relacionados a grade curricular ou temas diversos.

Para a criação de protótipos robóticos, podem ser utilizados *kits* de robótica, que são

conjuntos de peças específicas, como motores e sensores, que, utilizados juntamente com outras peças para a criação da parte mecânica, permitem a montagem de robôs. Segundo pesquisa que realizamos *in loco* durante a Olimpíada Brasileira de Robótica de 2012 [OBR 2012], os *kits* de robótica mais utilizados são os *kits* importados, que apresentam alto custo. Por esse motivo, existem iniciativas que buscam novas alternativas para substituir a utilização dos *kits* importados. Dentre estas iniciativas, podemos citar os *kits* de robótica com sucata e os simuladores robóticos.

Simuladores são *softwares* capazes de reproduzir o comportamento de algum sistema, produzindo fenômenos e sensações que na verdade não estão ocorrendo [Pedrosa 2010]. Os simuladores de robôs são capazes de simular os movimentos dos robôs e de reproduzir respostas similares aos dos sensores que os robôs possuem.

Neste trabalho propomos um simulador voltado para a robótica educacional que permite a realização de uma aula de robótica educacional sem a utilização de *kits* de robótica. Para a implementação do simulador robótico, é utilizada a linguagem de programação *Java* através da plataforma *NetBeans*. Para a implementação dos ambientes gráficos, é utilizado o *JMonkey Engine* [JMonkeyEngine 2002], um *game engine* que facilita substancialmente a construção de ambientes tridimensionais. O *JMonkey Engine* trabalha com recursos gráficos do *OpenGL* (*Open Graphics Library*) [OpenGL 2012]. Mais detalhes sobre as ferramentas utilizadas serão apresentados adiante neste texto, na Seção 2.4.

O robô virtual pode ser programado para seguir um conjunto de instruções definidas pelo programador. A programação do robô virtual é feita utilizando o *software* de programação R-Educ, desenvolvido pelos pesquisadores do laboratório Natalnet [NatalNet 2012], da Universidade Federal do Rio Grande do Norte. Embora a utilização do simulador e do *software* R-Educ seja paralela, os dois *softwares* estão diretamente ligados, ou seja, os programas desenvolvidos no R-Educ podem ser utilizados no simulador.

Foram realizados testes com relação ao funcionamento do simulador desenvolvido, e seus resultados serão discutidos no Capítulo 5. Dentre os resultados do trabalho, também estão incluídos os trabalhos científicos publicados acerca do simulador, que mostram o valor acadêmico deste projeto [Fernandes et al. 2012a, Fernandes et al. 2012b, Fernandes et al. 2012c].

1.1 Escopo do Trabalho

Visando aumentar a inserção tecnológica nas escolas públicas do Rio Grande do Norte, o Laboratório Natalnet vem desenvolvendo, desde 2003, vários projetos de Robótica Educacional. A partir de 2005, a equipe do Natalnet executou um projeto de pesquisa

com fomento do CNPq, no qual foi desenvolvida e colocada em prática uma metodologia para Robótica Educacional na Escola Municipal Prof. Ascendino de Almeida, localizada na periferia de Natal/RN, que finalizou em 2008. Para o desenvolvimento das aulas práticas foram utilizados *kits* de robótica Lego RCX e um *software* de programação, na época ainda em desenvolvimento, chamado R-Educ. Silva (2009) descreve os resultados dos testes realizados neste projeto, frisando que foram obtidos resultados satisfatórios com relação ao aprendizado dos alunos, embora o projeto não tenha tido continuidade na mesma escola, devido ao desinteresse dos professores envolvidos. De fato, o desenvolvimento do *software* R-Educ foi continuado por Barros (2008), que além de implementar cinco diferentes níveis de programação no R-Educ, realizou testes de usabilidade com esta ferramenta [Barros 2011].

A equipe envolvida no desenvolvimento do *software* R-Educ e na realização dos testes na Escola Ascendino de Almeida também desenvolveu outras ferramentas para serem utilizadas em aulas de robótica educacional. Levando em consideração a dificuldade de aquisição de *kits* de robótica, como, por exemplo, os *kits* Lego, esta equipe desenvolveu ferramentas capazes de aumentar a adesão das escolas às aulas de robótica educacional. Em um trabalho que foi o precursor deste [Fernandes 2010], nós desenvolvemos um simulador robótico bidimensional que permite a criação e execução de aulas de robótica educacional sem a utilização de *kits* de robótica. O simulador descrito neste trabalho é uma continuação do trabalho desenvolvido em 2010, buscando aprimorar a ferramenta (com inserção de visualização 3D), almejando sua melhor utilização e uma maior difusão do simulador.

Nesta mesma linha de trabalho (barateamento da Robótica Educacional), Sá (2011) criou um *kit* de robótica usando materiais de sucata, composto por motores, sensores e outras peças de sucata, que possui um custo total bastante inferior aos *kits* comerciais (menos de R\$ 100,00). De acordo com Sá (2011), os testes realizados com o *kit* tiveram resultados satisfatórios, mostrando que ele pode ser utilizado em aulas de robótica educacional.

É importante ressaltar que os trabalhos da nossa equipe visam que uma maior quantidade de alunos de ensino fundamental possa ter contato com um ambiente de aprendizagem tão rico e motivador quanto o ambiente da robótica educacional.

1.2 Motivação

De acordo com Neves-Júnior (2011), para o desenvolvimento de uma aula de robótica educacional são necessários os seguintes recursos em sala de aula:

- Um *kit* de robótica, composto por controlador lógico programável, sensores, motores e peças para construção de uma estrutura mecânica para a carcaça do robô;
- Um *software* de desenvolvimento de programas;
- Um computador capaz de utilizar o *software* escolhido; e
- Um ambiente propício ao desenvolvimento de uma determinada atividade.

Para a escolha dos equipamentos citados acima é necessário levar em consideração a faixa etária dos alunos que participarão da aula de robótica educacional, para que o *kit* de robótica e o *software* de programação sejam adequados a eles, ou seja, não sejam muito complexos. Além disso, é necessário levar em consideração também o preço desses equipamentos.

Por exemplo, Silva (2009) desenvolve uma pesquisa em que são realizadas aulas de robótica educacional com alunos de uma escola pública do Rio Grande do Norte, tendo optado por utilizar o *kit* *Lego Mindstorms*, por este permitir uma facilidade de manuseio por crianças, exigindo, portanto, um menor tempo na montagem do robô, já que não utiliza parafusos ou porcas. Ainda, a escolha também leva em consideração o fato de que não é necessário o conhecimento de conceitos científicos como eletricidade, transistores, etc. Já Pinto (2011) descreve a utilização de outros *kits* de robótica, incluindo uso de sucata, em que são usados materiais alternativos em aulas de robótica com professores e instrutores de informática. Metodologias de Robótica educacional com sucata são, em sua maioria, mais complexas do que com uso de *kits* comerciais. Quando são utilizados esses *kits*, o tempo demandado na montagem do robô é maior, pois os alunos terão que buscar sucatas de equipamentos adequados, incluindo impressoras, discos rígidos, entre outros, e modificá-los, tornando-os apropriados para seu propósito de uso no robô.

Santos et al. (2010) desenvolveu uma pesquisa comparando os principais *kits* educacionais proprietários do mercado com *kits* de robótica livre. O resultado da pesquisa mostrou que, embora os *kits* de robótica livre possuam um custo menor, eles não possuem facilidade de manuseio do equipamento. Segundo Sá (2011), os *kits* de projetos de sucata possuem custos entre R\$50,00 e R\$400,00, enquanto que os *kits* comerciais podem chegar até a mais de R\$2.000,00. No entanto, os *kits* não-comerciais, em sua maioria, possuem limitações quanto a programação e a proteção dos componentes elétricos, e são adequados a faixas etárias mais altas.

Há que se ressaltar que o acesso aos *kits* de robótica no Brasil é dificultado porque não há uma boa relação entre recursos e custos. Os *kits* importados são, em sua maioria, mais caros, enquanto que os nacionais apresentam limitações de *hardware* e *software*. Essas limitações tornam restrita a inserção da robótica educacional a poucas escolas [Miranda et al. 2010, Filho & Gonçalves 2008].

Com o anseio de ajudar a inserir técnicas de ensino mais modernas nas escolas, possibilitando ao aluno o contato com novas tecnologias e novas metodologias de ensino, propomos a criação de um ambiente de robótica educacional simulado, chamado S-Educ. Neste ambiente, o aluno teria acesso a diferentes modelos de robôs e de sensores virtuais, que interagiriam com os outros elementos do ambiente simulado para desempenhar uma atividade proposta.

Em nossa proposta, o aluno é capaz de montar o robô virtual a partir de blocos já definidos, programá-lo, e testar o seu funcionamento no ambiente virtual criado pelo professor.

1.3 Metodologia

Para o desenvolvimento deste trabalho, dividimos o mesmo em três fases principais. Na primeira fase foi realizada uma pesquisa sobre as tecnologias de simulação robótica existentes no mercado e uma análise das funcionalidades necessárias em um simulador, assim como uma pesquisa sobre as tecnologias existentes para a implementação de tal ferramenta. Nesta fase do projeto, foi definida a utilização de Java e do *JMonkey Engine*.

A segunda fase trata da implementação do simulador S-Educ, utilizando os dados obtidos na fase anterior. Esta fase foi a mais longa de todo o projeto. Durante esta fase, não foram realizados testes de usabilidade.

Na terceira fase, de testes, foram realizados experimentos a respeito da utilização do simulador robótico desenvolvido em aulas de robótica educacional. Os testes envolveram professores de nível fundamental ou médio, professores de robótica educacional, alunos com conhecimento de robótica, alunos sem conhecimento de robótica, e até mesmo adultos leigos. Nesta última fase, foram desenvolvidas atividades a serem realizadas pelos participantes, buscando testar todas as funcionalidades do simulador. Os resultados dos testes serão apresentados adiante, no Capítulo 5.

1.4 Aplicações

Neste trabalho foi desenvolvido um simulador robótico que pode ser utilizado em aulas de robótica educacional, chamado S-Educ. O simulador pode ou não ser utilizado em conjunto com *kits* de robótica, diminuindo o custo operacional das aulas de robótica, o que permite a difusão deste ambiente de aprendizagem em escolas brasileiras. O simulador S-Educ foi desenvolvido para simular robôs do tipo *Lego Mindstorms NXT* [LEGO 2012],

permitindo a simulação da movimentação dos motores e da detecção de dados através dos sensores de robôs criados com este *kit*.

Dentre as vantagens da utilização de um simulador, vale destacar a diminuição do custo das aulas e do tempo dos testes, e a facilidade de testar algoritmos. Além disso, é possível detectar previamente situações durante a execução da atividade que podem causar danos ao robô real. Por fim, os simuladores permitem que os alunos se familiarizem com a tecnologia, mesmo sem a presença física do robô, adquirindo os conhecimentos básicos necessários antes de utilizarem o robô real [Hoss et al. 2009].

Embora a utilização de simuladores apresente muitas vantagens ao usuário, também há desvantagens relacionadas ao uso do simulador em Robótica Educacional. Entre as desvantagens, destacamos a montagem do robô, que é feita virtualmente. A montagem de um robô é um passo de suma importância em uma aula de robótica educacional, pois permite que o aluno estimule a criatividade e principalmente a coordenação motora. Este passo não é executado quando são utilizados simuladores, e por isso estimula-se a utilização de simuladores juntamente com *kits* de robótica.

Uma aula de robótica educacional utilizando o S-Educ pode possuir duas vertentes: aula com simulador e aula com simulador e *kit* de robótica. No primeiro caso, os alunos utilizam apenas o simulador durante a aula de robótica, sendo todos os testes realizados no ambiente virtual. Neste caso o aluno não tem contato com nenhum *kit* de robótica educacional. O custo deste tipo de aula é ainda mais reduzido devido à diminuição dos custos com materiais de aula, como cartolinhas, fita isolantes, etc.

No segundo caso, os alunos utilizariam o simulador e um *kit* de robótica educacional. Este tipo de configuração de aula só é possível porque o mesmo programa desenvolvido para o simulador S-Educ pode também ser utilizado para programar o robô real (o *software* R-Educ é utilizado). Note que o simulador pode ser utilizado para a realização de experimentos, diminuindo o custo com pilhas e baterias e agilizando os testes. Depois que o algoritmo desenvolvido for testado, os alunos podem iniciar os testes no robô, efetivamente. Neste caso, pode ser que sejam necessários pequenos ajustes no programa para a conclusão da atividade. Neste tipo de aula, o custo com os *kits* pode ser diminuído se for diminuída a quantidade de *kits* adquiridos pela escola. Como os *kits* só serão utilizados no final da aula, e a maioria dos testes será feita no simulador, mais de um grupo podem dividir um só *kit*.

1.5 Estrutura do trabalho

Este trabalho introduz um simulador que pode ser utilizado em aulas de robótica educacional. No Capítulo 1 mostramos o escopo do trabalho, destacando os trabalhos desenvolvidos na área de robótica educacional relacionados a este simulador. Expomos uma visão geral da motivação deste trabalho, que é a criação de uma solução para o problema de implantação de aulas de robótica educacional em escolas brasileiras. Além disso, descrevemos a metodologia que foi utilizada e as aplicações deste tipo de tecnologia.

O Capítulo 2 apresenta as fundamentações teóricas relacionadas com este trabalho, como explicações sobre robótica educacional, simulação de robôs e *game engines*, em especial o *JMonkey Engine*, que foi o *game engine* utilizado para o desenvolvimento deste trabalho. No Capítulo 3, serão apresentados alguns simuladores de robótica educacional, fazendo um comparativo entre eles e o S-Educ.

O Capítulo 4 descreve detalhadamente o problema da simulação de robôs na robótica educacional, especificando como foram desenvolvidas as simulações da movimentação do robô virtual, de sensores e de atuadores. Além disso, exibe a interface gráfica da aplicação, assim como diagramas *UML* descrevendo o funcionamento do simulador e a forma de utilização de tal ferramenta.

No Capítulo 5 são expostos alguns exemplos de aulas de robótica que podem ser desenvolvidas utilizando o simulador S-Educ. São mostrados também os experimentos realizados com o simulador, separados pelos diferentes grupos de testes, e os resultados obtidos nestes experimentos. Por fim, no Capítulo 6 serão apresentadas as conclusões, formuladas a partir dos experimentos realizados.

Capítulo 2

Fundamentação Teórica

Como visto no Capítulo 1, neste trabalho desenvolvemos um simulador de robôs voltado para uso em robótica educacional. O simulador, chamado S-Educ, pode ser utilizado em escolas para suprir a falta de *kits* de robótica, possuindo muitas das funcionalidades de um robô, contribuindo para a modificação do modelo tradicional de aula, motivando o professor e o aluno.

Neste capítulo, colocamos todo o embasamento teórico necessário ao entendimento dos conceitos usados neste trabalho, que são as definições e aspectos metodológicos e práticos da robótica educacional, o uso de simuladores de robôs e a ferramenta para a criação de ambientes tridimensionais que foi utilizada no desenvolvimento do simulador.

2.1 A Robótica Educacional

As transformações dos sistemas educacionais são, em sua maioria, mais lentas do que as transformações no mundo tecnológico. Por esse motivo, as escolas ainda estão formando alunos sem fluência digital, ou seja, alunos que tem conhecimento sobre as tecnologias, mas não sabem utilizá-las [Neves-Júnior 2011].

Busca-se inserir novas tecnologias no âmbito acadêmico, estabelecendo uma melhor qualidade de ensino e ambientes de aprendizagem mais ricos e motivadores para os alunos [Miranda 2006]. Dentre estas tecnologias encontra-se a robótica educacional. Segundo o Dicionário Interativo da Educação Brasileira [DIEB 2012], o termo robótica educacional pode ser definido como:

Termo utilizado para caracterizar ambientes de aprendizagem que reúnem materiais de sucata ou kits de montagem compostos por peças diversas, motores e sensores controláveis por computador e softwares que permitam programar de alguma forma o funcionamento dos modelos montados.

A robótica educacional caracteriza-se como um ambiente em que os alunos têm a opção de montar e programar seu próprio sistema robótico, por meio de observações e da própria prática [Fortes 2007, Silva 2009].

As aulas de robótica educacional podem envolver diversos temas, desde temas sobre ética e responsabilidade social até temas dos tradicionais componentes curriculares, sempre trabalhando de forma lúdica e atrativa aos alunos. Estas aulas possibilitam que os problemas colocados pelo professor sejam discutidos em grupo, incentivando assim o trabalho colaborativo para a obtenção das devidas soluções. Tudo isso a partir de uma interação entre alunos, professores e recursos tecnológicos de *hardware* e de *software*. A Figura 2.1 mostra um exemplo típico de uma aula usando recursos de robótica educacional, na qual os alunos montaram um robô e o programaram para participar de uma competição em tema versando sobre os estados do Brasil.



Figura 2.1: Exemplo de uma aula de robótica educacional.

Zilli (2004) menciona algumas competências que podem ser desenvolvidas no aluno mediante sua participação em aulas de robótica educacional, como o raciocínio lógico, a investigação e a compreensão, o trabalho com pesquisa e a utilização da criatividade em diferentes situações.

Uma aula de robótica educacional pode ser separada em quatro momentos distintos [Miranda 2006]:

1. **A problematização:** é o momento de apresentação da atividade, que sempre está relacionada a uma situação-problema instigante para os alunos. A situação-problema pode envolver temas interdisciplinares ou sociais, acessíveis aos alunos, depen-

dendo de sua faixa etária. Podem ser introduzidos novos conceitos aos estudantes, desde que não seja muito discrepante ao seu nível de conhecimento;

2. **Exploração de potenciais soluções:** o professor incentiva um diálogo e interação entre os alunos, criando discussões que auxiliem o grupo na elaboração de estratégias para resolução do problema. O grupo deve utilizar as explicações do professor na aula, assim como o seu conhecimento prévio sobre robótica e programação;
3. **Desenvolvimento das soluções:** a partir das soluções encontradas no segundo momento, o grupo deve montar um robô capaz de resolver a situação-problema. Em seguida, eles devem criar um programa seguindo os passos escolhidos no segundo momento. A criação do programa pode ser desenvolvida em qualquer *software* de programação de robôs; e
4. **Análise dos resultados:** o grupo realiza testes para verificar se o robô montado e programado consegue resolver a situação-problema. Caso a montagem ou a programação do robô necessitem de ajustes, o grupo deve voltar para o terceiro momento. Caso a solução escolhida no segundo momento não apresente bons resultados, o grupo deve retornar ao segundo momento e iniciar uma nova discussão sobre possíveis soluções.

Sem dúvida, o pioneiro no desenvolvimento de metodologias para uso da robótica com fins educacionais foi Seymour Papert, pesquisador do Instituto de Tecnologia de Massachusetts (MIT) [Papert 1985]. Papert desenvolveu uma linguagem de programação chamada LOGO, inicialmente utilizada para programar uma tartaruga mostrada apenas de forma gráfica na tela de um computador. Posteriormente, Papert desenvolveu uma tartaruga mecânica, chamada de *tartaruga de solo*, para possibilitar uma representação mais concreta para crianças. Esta iniciativa foi pioneira na utilização da robótica na educação e inspirou os trabalhos posteriores [Gonçalves 2007].

2.2 Ferramentas utilizadas em aulas de robótica educacional

Como visto no Capítulo 1, em uma aula de robótica educacional há a necessidade de utilização de *hardware* e *software*. O *hardware* refere-se a um *kit* de robótica para construção do robô, enquanto que o *software* envolve uma *ferramenta* para a programação do robô. *Kits* de robótica são conjuntos de peças específicas que, juntas, são capazes de formar as partes físicas dos robôs. Os *kits* podem ou não ser programáveis. *Kits* programáveis são aqueles em que os usuários podem desenvolver diferentes programas para modificar o

comportamento dos robôs. Já os *kits* não programáveis são aqueles em que o robô sempre executará as mesmas instruções. Em ambos os casos, os kits podem ser equipados com motores, sensores e peças diversas para a construção do robô.

Um *software* de programação é um programa de computador que permite que um aluno consiga programar comandos capazes de lidar com os atuadores (motores) e sensores de um robô, permitindo criar diferentes tipos de programação para diferentes tipos de robôs. Embora Neves-Júnior (2011) tenha citado apenas os *softwares* de programação, existem também os *softwares* de simulação, que podem ser utilizados em aulas de robótica educacional para simular a programação dos robôs e/ou até mesmo a sua montagem.

2.2.1 *Hardware*

Diferentes tipos de *hardware* (*kits* de robótica) podem ser usados em uma aula de robótica educacional, podendo eles serem classificados em três categorias principais, a saber, os *kits* comerciais, os *kits* de sucata e os *kits* com componentes comerciais.

A primeira categoria envolve os *kits* de robótica comerciais, como Lego NXT e VEX [VEX 2012]. A segunda categoria, *kits* com sucata, são *kits* de robótica nos quais os materiais que montam a carcaça do robô são feitos de sucata, como, por exemplo, o *kit* desenvolvido por Sá (2011). Na maioria dos casos, é utilizada sucata tecnológica, tais como dispositivos leitores de discos diversos (CD, *floppy*, discos rígidos antigos), impressoras, computadores ou aparelhos de som. Por fim, na terceira categoria, os *kits* usados são compostos apenas por componentes eletrônicos e peças comercialmente disponíveis. Nesta categoria, podemos citar como exemplo o *kit* N-Bot, desenvolvido por Aroca et al. (2012), no qual são usados componentes eletrônicos para fazer a parte de controle de baixo nível do robô (para ler sensores e mover atuadores) e peças (novas) que são geralmente utilizadas na fabricação de brinquedos (polias, rodas, engrenagens, eixos) e de móveis (rodas bobas, parafusos, suportes de metal), que podem ser facilmente adquiridas em lojas comerciais.

Os *kits* comerciais apresentam, em sua maioria, uma melhor qualidade quando comparados aos *kits* das outras categorias [Miranda et al. 2010]. No entanto, isso vem atrelado a um alto custo do produto. Este fator pode tornar esses equipamentos inacessíveis a determinados públicos.

Os *kits* de sucata são oriundos, na maior parte dos casos, de pesquisas em universidades e, em menor ocorrência, de algumas poucas iniciativas, isoladas, em escolas públicas. Nas pesquisas, busca-se desenvolver um *kit* de robótica com a mesma qualidade dos *kits* comerciais e a um custo menor. No entanto, grande parte destes projetos possuem limita-

ções de *hardware*, oriundos da pequena quantidade de testes, ou da utilização de materiais de baixa qualidade. Por esse motivo, nem todos os *kits* de robótica com sucata possuem os requisitos necessários para serem utilizados em aulas de robótica educacional [Miranda et al. 2010, Gonçalves 2007]. Aliando-se a isso uma rejeição natural do ser humano, por se tratar de sucata, que quase sempre é entendida como sendo um sinônimo de lixo, esse tipo de kit não é muito disseminado entre as escolas, principalmente na rede privada de ensino.

Os *kits* baseados em componentes são *kits* que utilizam apenas componentes eletrônicos e peças disponíveis no comércio. Uma tentativa recente de criação de um *kit* baseado em componentes, de baixíssimo custo, visando levar a robótica a populações de baixa renda ocorreu com a criação da entidade denominada de **AFRON** (*African Robot Network*) que promoveu a competição *AFRON 10-dollar robot design challenge*, com a intenção de fomentar o desenvolvimento de plataformas robóticas abertas de baixíssimo custo, especialmente desenvolvidas para educação [AFRON 2012]. Embora similares aos *kits* de sucata, esses *kits* não utilizam nenhum material reciclável e são mais eficazes que aqueles, pois geralmente os componentes usados são específicos para tal finalidade. Ou seja, o fato de não utilizar materiais de sucata é uma vantagem deste *kit*, pois aumenta a qualidade do robô criado, embora aumente um pouco o custo e também a complexidade do *kit*, pois o usuário necessita de um conhecimento sobre eletrônica para o uso de tal ferramenta.

Convém ressaltar que, embora existam várias alternativas para os *kits* de robótica, poucos apresentam uma boa relação entre custo e facilidade de uso. Este problema dificulta a adesão da robótica educacional nas escolas brasileiras.

2.2.2 *Software*

Existem vários *softwares* que podem ser usados para a programação de robôs, também conhecidos como ambientes integrados de desenvolvimento (ou *IDE* do inglês *integrated development environment*) ou ambientes gráficos de programação (ou *GPE* do inglês *Graphical Programming Environment*). No contexto deste trabalho, os *softwares* utilizados em aulas de robótica educacional podem se encaixar, basicamente, em duas categorias, os *Softwares* de programação e os *Softwares* de simulação.

Os *softwares* de programação são os programas de computador que utilizam uma linguagem de programação para programar os movimentos de um robô. Este tipo de *software* pode ser utilizado para programar robôs reais, montados a partir de *kits* de robótica, ou robôs simulados. Por si só, o *software* não é suficiente para a realização de uma aula de ro-

bótica educacional. O NXT-G é um GPE capaz de programar robôs Lego NXT, utilizando a linguagem de programação também chamada de NXT-G, que é uma linguagem visual, ou gráfica. O BricxCC [Brickxcc 2012] é um IDE que utiliza a linguagem de programação NXC [NXC 2012] para programar robôs Lego NXT, mas podendo usar outras linguagens, e outros robôs, como o Lego RCX. A linguagem R-Educ [Barros 2011, Sá 2011], com IDE de mesmo nome, foi desenvolvida por pesquisadores do Laboratório Natalnet para programar robôs Lego RCX, Lego NXT ou a plataforma H-Educ [Sá 2011]. A metodologia RoboEduc [Silva 2009] utiliza esta linguagem, que é organizada em níveis de abstração, dependendo do público alvo e da plataforma de *hardware* utilizada.

A segunda categoria envolve os simuladores de robôs, que são programas capazes de simular o comportamento de robôs em um ambiente virtual que simula o ambiente real no qual ocorreria uma aula de robótica educacional. Para a utilização de *softwares* de simulação também é necessário o uso de *softwares* de programação, que permitirão a programação do robô virtual. Em alguns casos, a programação é feita diretamente no *software* de simulação.

Neste trabalho desenvolvemos um simulador para ser utilizado em aulas de robótica educacional, buscando difundir este ambiente de aprendizagem. Todos os momentos descritos acima presentes em uma aula de robótica podem ser feitos no ambiente virtual. O aluno é capaz de montar, programar e testar todos os passos do seu projeto. Na Seção 2.3, a seguir, são descritas as características de um simulador de robôs.

2.3 Simuladores Robóticos

No modelo geralmente adotado para o desenvolvimento de atividades com robôs, um grupo de testes que é composto por construtores e programadores de robôs, planeja e realiza um conjunto de testes com relação à montagem e à programação dos mesmos, observando e analisando o resultado de cada teste. O ideal é que estes testes sejam realizados em um robô real, mas isso pode ser um pouco complicado, principalmente quando é necessário mais de um robô [Pedrosa 2010].

Dentre os possíveis impasses para a utilização do robô real, podemos citar:

- O grupo de testes não possui uma quantidade suficiente de robôs;
- Os robôs podem ser danificados pelos testes;
- O espaço físico disponível para os testes pode não ser suficientemente grande; e
- O grupo de testes pode não estar perto do robô.

Para diminuir o custo e o tempo de pesquisa no desenvolvimento de robôs, várias

pesquisas desenvolvem e usam ferramentas próprias para prover uma maneira fácil e simples de testar ideias, teorias e programas com robôs sem depender fisicamente da máquina. A maneira mais simples de fazer isso é a partir de simuladores computacionais [Becker 2010, Obst & Rollmann 2004]. Simuladores são ambientes computacionais que emulam o acontecimento de algum fenômeno real que os usuários conseguem manipular, explorar e experimentar [Jonassen 1996].

Em algumas experiências, os testes se tornam extensos, sendo necessária a constante repetição dos testes com os robôs. Em outros casos, são necessárias várias modificação no modelo do robô, ou seja, o uso de diferentes tipos de carcaças, bases e sensores. A adaptação do robô e a contínua repetição dos testes custariam um tempo muito grande do grupo de testes, sem mencionar a troca de baterias da máquina.

Um simulador robótico é capaz de modelar ambientes que simulam o ambiente real no qual o robô está inserido, e robôs que simulam o comportamento do robô real. Os simuladores permitem que o usuário visualize e interaja em tempo real com o ambiente virtual, de forma que o usuário possa realizar as mesmas atividades que realizaria no ambiente real.

O ambiente virtual de simulação robótica é uma ferramenta poderosa que apresenta uma série de vantagens [Wolf et al. 2009, Michel 2004, Pedrosa 2010]:

- **Economia de recursos financeiros:** o custo de um *software* de simulação é menor do que o custo de *kits* de robótica comerciais. Além disso, quando se está utilizando simuladores não é necessária a compra de equipamentos para a utilização no ambiente no qual o robô ficará, já que todos estes equipamentos serão virtuais;
- **Facilidade de criação do ambiente que será utilizado pelo robô:** a criação do ambiente real exige a utilização de equipamentos, o que demanda tempo para a organização de tais equipamentos na sala. A criação do ambiente virtual consome um tempo menor, já que não é necessário mover equipamentos, apenas adicioná-los no ambiente virtual;
- **Economia de tempo:** em ambientes simulados há uma economia de tempo na realização de atividades. Dentre os principais fatores, podemos citar a facilidade de criação do ambiente que será utilizado pelo robô e a facilidade de reconfiguração do ambiente para novos testes (repositionar robôs e equipamentos). Isso permite que, em um mesmo intervalo de tempo, o ambiente virtual permita a realização de um maior número de experimentos com o robô.
- **Utilização de robôs de maior qualidade:** os robôs sofrem desgastes com o tempo, o que faz com que o seu funcionamento varie com o tempo. Por exemplo, motores antigos não se comportam da mesma forma que motores novos. Robôs simulados

não possuem desgastes em suas peças, e por isso não há erros inesperados nos testes, oriundos do desgaste das peças do robô;

- **Evita danos ao robô:** através das simulações pode-se verificar previamente as situações que podem provocar danos ao robô, devido, por exemplo, a fortes colisões ou exposição do robô a ambientes perigosos para testes;
- **Economiza bateria dos robôs:** o tempo de vida da bateria pode ser aumentado pela utilização do simulador;
- **Facilita o teste de novos algoritmos e modelos de robôs:** devido à economia de tempo na reconfiguração do ambiente de testes, os testes são bem mais simples no ambiente virtual; e
- **Facilita o teste com vários robôs:** este tipo de teste seria mais difícil no ambiente real, já que o grupo de testes deveria possuir vários robôs.

Caso o grupo de testes tenha acesso a um modelo real do robô virtual que está sendo utilizado, eles podem também testar as aplicações desenvolvidas no ambiente virtual em um ambiente real, com mínimas modificações. Assim, o ambiente virtual seria utilizado para realizar a maior parte dos testes.

Este tipo de tecnologia já está sendo utilizado em algumas competições de robótica, como a *RoboCup* [RoboCup 2012]. Nesta competição existem quatro categoriais que competem em ambientes simulados:

1. *Robocup Soccer 2D Simulation League*: competição de futebol;
2. *Robocup Soccer 3D Simulation League*: competição de futebol com robôs do tipo humanóide;
3. *Rescue Simulation League*: uma equipe de robôs virtuais deve percorrer o ambiente virtual para resgatar vítimas; e
4. *RoboCupJunior - CoSpace Demo Challenge*: possui dois tipos de competições diferentes: aventura e dança.

2.3.1 Simuladores Robóticos na Educação

Em aulas de robótica educacional utilizando simuladores robóticos, o simulador pode ser utilizado para atrair os alunos a um ambiente lúdico, facilitando o ensino de temas interdisciplinares.

Encontram-se na literatura algumas iniciativas de desenvolvimento de ambientes de programação específicos para a área educacional e utilizados em aulas de robótica educacional. Para o uso de simuladores robóticos na educação é necessário que os alunos

tenham acesso a um computador, um robô virtual (composto por motores, sensores e peças para carcaça), um *software* de programação e um ambiente virtual onde se possa executar a aula de robótica educacional.

Embora apresentem diversas vantagens quando comparados aos robôs reais, como citado na Seção 2.3, os simuladores não conseguem substituir por completo a utilização de *kits* de robótica. A montagem de robôs reais é uma forma lúdica de aperfeiçoar a coordenação motora, além de ser mais atrativa aos alunos do que a montagem de um robô virtual. Além disso, nem todos os simuladores são capazes de simular todos os movimentos desenvolvidos por um robô.

Mesmo apresentando algumas desvantagens, os simuladores podem ser utilizados com sucesso em aulas de robótica educacional sem a utilização de robôs. Alguns trabalhos, como os de Benitti et al. (2009) e Gomes et al. (2008), citam o uso de simuladores robóticos em aulas de matemática e geografia. Os autores citam o estímulo dos alunos no desenvolvimento do conhecimento e um ganho em conhecimento sob o aspecto tecnológico, de inclusão social e digital, e também um estímulo dos professores com relação ao desenvolvimento das aulas utilizando simuladores robóticos para participar de seu cotidiano. Os detalhes destes trabalhos, assim como outros simuladores robóticos, serão mostrados no Capítulo 3.

2.4 Tecnologias Utilizadas

Como descrito no Capítulo 1, o simulador proposto foi implementado em *Java*, com o uso do *Game Engine 3D JMonkey Engine*. *Games Engines* são sistemas ou conjuntos de bibliotecas utilizados com o objetivo de simplificar e abstrair o desenvolvimento de jogos ou aplicações com gráficos. Eles permitem renderizar gráficos bidimensionais ou tridimensionais, simular elementos físicos, detectar colisão, além de suportar animação, sons, inteligência artificial, *networking*, entre outros [Bisolo & Bughi 2009].

O *Game Engine* utilizado neste projeto foi o *JMonkey Engine* (JME). O JME é baseado em *Java*, gratuito e multiplataforma, baseado em OpenGL, e permite o desenvolvimento de aplicativos para computador, internet ou plataformas móveis. O JMonkey Engine possui alta performance e uma comunidade ativa, criando uma API em constante evolução. Além disso, possibilita uma alta velocidade de renderização, tem vários recursos específicos que aceleram a implementação, e suporta múltiplas colisões por objeto [Araujo 2010, Steinbauer 2007, Guzinski 2009].

O JME permite a captura da imagem do ambiente virtual, além da criação de vídeos. Além disso, possui estruturas de dados pré-definidas que permitem a criação de objetos

de vários tipos, como veículos, objetos imóveis e personagens. A utilização deste tipo de estruturas de dados facilitam a criação de objetos virtuais, utilizando funções e características já definidas.

Foi realizada uma pesquisa para avaliar qual seria a melhor opção de biblioteca 3D para ser utilizada no desenvolvimento desta ferramenta. Levando em consideração os resultados obtidos por alguns trabalhos científicos [Guzinski & Maillard 2009, Cogley et al. 2008, Steinbauer 2007], decidimos utilizar a biblioteca *JMonkey Engine*, visto que esta possui uma execução rápida, e funções de mais alto nível, específicas, que agilizam o processo de implementação da arquitetura.

Com relação aos modelos tridimensionais das peças dos robôs, utilizamos alguns modelos prontos, disponíveis no Armazém 3D do Google [Google 2012]. Para modificar esses modelos, ou criar modelos mais complexos a partir deles, utilizamos o Google SketchUp [SketchUp 2012], um *software* livre para a criação de modelos 3D. Escolhemos este *software* devido ao conhecimento que tínhamos com a ferramenta.

Capítulo 3

Trabalhos Relacionados

Os simuladores robóticos são ferramentas que facilitam a utilização e os testes de robôs em diversas aplicações, tal como em cirurgias médicas, em indústrias e em robótica educacional.

Neste capítulo apresentamos alguns simuladores robóticos do mercado voltados para educação, e em seguida realizamos uma comparação entre os simuladores apresentados e o simulador S-Educ, aqui proposto, enfatizando neste comparativo as ferramentas de programação e montagem dos robôs virtuais.

3.1 Simuladores para Robótica Educacional

Os simuladores voltados para a robótica educacional podem ser divididos em três categorias: simuladores para montagem de robôs, simuladores para a programação de robôs, e simuladores para a montagem e programação de robôs. A escolha da categoria depende do tipo de atividade que se deseja realizar.

3.1.1 Simuladores para montagem de robôs

Alguns simuladores focam apenas na criação do robô virtual, sendo que não é possível realizar testes quanto ao funcionamento do mesmo. A experiência com este tipo de programa consiste em atividades que envolvem a aprendizagem de transformações geométricas por parte dos alunos.

O *Lego Digital Designer* (LDD) [LDD 2012] é um programa que permite a manipulação e montagem de peças virtuais do *kit* de robótica da *Lego*, o *Mindstorms*. O simulador possui uma grande quantidade de peças dos mais variados tamanhos e formatos. No entanto, este *software* não permite que o usuário exporte o modelo 3D criado, limitando o uso do robô virtual a esta ferramenta.

Além do LDD, alguns *softwares* comerciais de modelagem 3D, como *Autodesk Inventor* [Autodesk 2012] e *Blender* [Blender 2012], podem ser utilizados para simular a montagem de um robô. Para tal, basta que o usuário faça o *download* de peças de robôs, de forma que o *software* seja usado para conectá-las. Estes *softwares* não são propriamente considerados como simuladores robóticos, visto que possuem diversas outras características, sendo capazes de criar outros tipos de modelos tridimensionais além de robôs, mas foram mencionados apenas por serem *softwares* capazes de permitir a montagem virtual de robôs.

3.1.2 Simuladores para programação de robôs

Há simuladores que permitem a programação de modelos robóticos virtuais, mas não possibilitam ao usuário a montagem do robô virtual. O usuário fica limitado a utilizar os modelos disponíveis no programa ou fazer o *download* de novos modelos (caso o simulador permita esta opção). Dentre os simuladores desta categoria destacamos o *RoboMind* [RoboMind 2012], o *KickRobot* [Gomes et al. 2008] e o ambiente de simulação 2D desenvolvido anteriormente por nós para a metodologia RoboEduc [Fernandes 2010].

RoboMind

O *RoboMind* [RoboMind 2012] é um programa que oferece uma linguagem de programação simples para programação de um robô em um mundo bidimensional, conforme ilustra a Figura 3.1(a), que apresenta a tela principal do ambiente virtual do *RoboMind*. O robô se movimenta através da programação do usuário, que indica a quantidade de casas que ele deve andar.

O robô pode ser equipado com apenas um sensor ultrassônico que pode ver se há objetos na frente ou nos lados do robô. Com relação aos atuadores, o robô pode possuir um atuador capaz de segurar e carregar objetos e um atuador capaz de pintar a casa na qual o robô está localizado.

O ambiente virtual é editável e pode ser salvo em arquivos de texto. Atualmente, o *RoboMind* está disponível em oito idiomas, incluindo português do Brasil.

Benitti et al. (2009) realizou atividades de matemática e geografia utilizando o simulador *RoboMind*, e constatou um aumento no número de acertos em perguntas específicas, além de um aumento no conhecimento específico tecnológico.

KickRobot

O *KickRobot* [Gomes et al. 2008] (Figura 3.1(b)) é um simulador no qual os alunos utilizam robôs em uma competição virtual de basquete. O simulador possui apenas um robô capaz de arremessar bolas em uma cesta virtual.

O objetivo da utilização deste simulador é a aplicação de teoremas matemáticos e fórmulas físicas na realização da atividade. Neste simulador o robô é fixo, não possui sensores, e possui apenas um atuador, capaz de arremessar as bolas. Gomes et al. (2008) destaca o ganho em conhecimento adquirido com este simulador, além do alto índice de interesse dos alunos na atividade.

Ambiente de simulação 2D (RoboEduc)

O ambiente de simulação, em sua versão 2D original usada na metodologia RoboEduc [Fernandes 2010], foi desenvolvido com o propósito simples de auxiliar em aulas de robótica para alunos do ensino fundamental. Para a programação do robô virtual, deve ser utilizado o *software* educacional R-Educ. O simulador é bidimensional e possui apenas um tipo de robô, não permitindo a modelagem de novos protótipos robóticos.

O ambiente do mesmo, mostrado na Figura 3.1(c), pode ser modificado com a adição de novos objetos, que são sempre do mesmo tamanho, mas podendo ter cores diversas. Embora possam ser reconhecidos pelo próprio robô ou por sensores, estes objetos não podem ser carregados pelo robô, ou seja, este robô não possui nenhum manipulador, que são atuadores que permitem manipular objetos. O robô possui sensores de toque, cor e/ou ultrassônico. Como característica marcante, desejável, o programa desenvolvido pode ser enviado a um robô real do tipo *Lego Mindstorms NXT*, *Lego RCX* ou H-Educ, que executará o programa sem problemas de execução (possivelmente batendo em algum objeto do ambiente real, claro, mas executando as instruções do programa).

Utilização em aulas de robótica educacional

Os simuladores desta categoria, *RoboMind*, *KickRobot* e o Ambiente de simulação (2D) original da RoboEduc, são simuladores que podem ser utilizados em aulas de robótica educacional. Benitti et al. (2009), Gomes et al. (2008) e Fernandes (2010) citam exemplos de aulas de matemática, geografia ou lógica que podem ser realizadas utilizando esses ambientes de simulação. Como característica comum, esses três simuladores possuem visualizações bidimensionais do ambiente virtual, o que os torna mais alusivos às crianças de ensino fundamental.

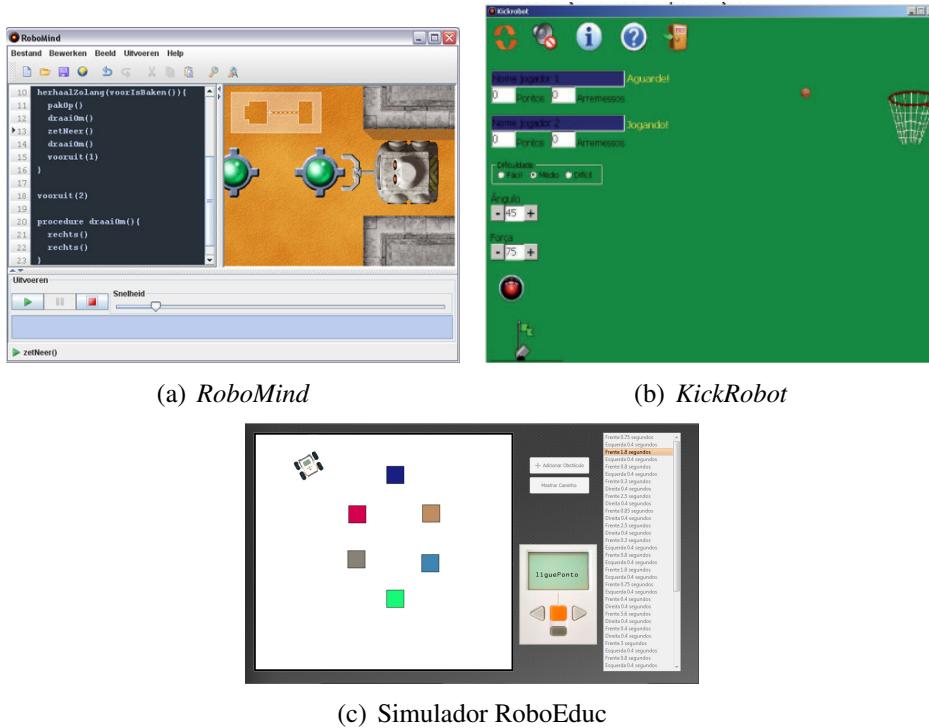


Figura 3.1: Simuladores para programação de robôs.

3.1.3 Simuladores para montagem e programação de robôs

Um simulador que permite a montagem e a programação de um robô virtual possibilita uma simulação ainda mais autêntica do ambiente real. A utilização desse tipo de simulador permite emular quase todos os passos de uma aula de robótica educacional que não utiliza simulador, desde a criação do modelo robótico até a sua utilização para a resolução de uma determinada situação problema. Dentre os simuladores desta categoria, vale destacar o *USARSim* [USARSim 2012], o *Robot Virtual Worlds* [RVW 2012] e o *Webots* [CyberBotics 2012].

USARSim

O *USARSim* (*Unified System for Automation and Robot Simulation*) [USARSim 2012] é um simulador de robôs de busca e resgate, criado como uma ferramenta de pesquisa. Este simulador inicialmente suportava apenas robôs com rodas, mas devido a um aumento na utilização desta ferramenta, as versões atuais suportam também robôs submarinos, humanoides e helicópteros. A Figura 3.2(a) mostra a tela do simulador *USARSim*.

Este simulador apresenta modelos de ambientes virtuais pré-definidos, simulando ambientes ao ar livre e ambientes internos. No entanto, é possível que o usuário desenvolva

novos ambientes facilmente.

O usuário tem a opção de criar o robô e o ambiente virtual. O robô pode possuir atuadores e alguns sensores, como sensor de posição, que indica rotação, localização e velocidade, e sensores de percepção, como o sensor ultrassônico ou de toque. Na criação do robô o usuário tem a opção de definir parâmetros como massa, fricção, velocidade e torque do motor.

O programa desenvolvido para o ambiente de simulação pode ser enviado para o robô real e vice-versa. Este ambiente de simulação é usado em competições de robótica, como a *RoboCup Rescue Simulation League*. Embora este tenha sido o objetivo inicial do desenvolvimento deste simulador, atualmente ele pode ser utilizado para um propósito maior, podendo ser expandido para outras aplicações.

Robot Virtual Worlds

O *Robot Virtual Worlds* [RVW 2012] é um ambiente que permite ao usuário utilizar modelos virtuais de robôs do tipo *Lego NXT*, Vex ou Tetrix em um ambiente virtual bidimensional ou tridimensional. O robô virtual pode possuir todos os sensores e motores que o *kit* de robótica real possui. O simulador, exibido na Figura 3.2(b), possui diversos tipos de ambientes virtuais pré-configurados, e permite que o professor da aula de robótica crie novos ambientes para serem utilizados pelos alunos.

A linguagem de programação usada para programar o robô virtual é a *RobotC*, e portanto o programa criado pode ser enviado para um robô real. No entanto, o simulador não apresenta esta opção, ou seja, o usuário precisa utilizar outro programa para isso.

Webots

O simulador Webots [CyberBotics 2012] é um simulador tridimensional desenvolvido pela Cyberbotics Ltda que permite a modelagem, programação e simulação de robôs móveis. O usuário pode adicionar objetos passivos (objetos da cena) ou ativos (robôs). Em uma só cena podem ser adicionados vários objetos, permitindo que dois robôs possam interagir entre si em um ambiente. Os robôs podem se locomover através de rodas, pernas ou asas. A Figura 3.2(c) mostra um exemplo de um robô modelado no Webots, para o qual pode ser desenvolvido um programa de controle.

Os robôs podem conter diversos tipos de sensores e atuadores, como sensores de distância e toque, câmeras, entre outros. Cada robô pode ser programado individualmente para realizar um determinado conjunto de ações.

Os programas que são utilizados para controlar o robô são chamados de controladores.

Esses controladores podem ser escritos nas seguintes linguagens: C, C++, Java, Python, URBI ou *MATLAB_{TM}*.

O simulador Webots permite que o programa gerado e simulado seja enviado para robôs reais, como Khepera, Hemisson, Lego Mindstorms, entre outros. Apesar de ser um dos simuladores mais avançados na sua área, o *Webots* não é muito usado em pesquisas devido ao alto custo.

Utilização em aulas de robótica educacional

Em comparação com os simuladores expostos na Seção 3.1.2, os simuladores *USARSim*, *Robot Virtual Worlds* e *Webots* são mais complexos, tanto com relação ao ambiente virtual quanto com relação aos robôs utilizados. Os ambientes virtuais podem ser das mais diversas formas, desde ambientes ao ar livre até casas com escadas. Os robôs não se limitam apenas a robôs com rodas, já que é possível a simulação de robôs com pernas, asas, ou com outros dispositivos de locomoção.

Embora apresentem complexidades maiores, estes simuladores podem ser utilizados em aulas de robótica educacional, mas por alunos com um nível maior de conhecimento sobre montagem e programação de robôs. O simulador *USARSim*, por exemplo, é utilizado em competições, por alunos de nível universitário.

3.2 Análise Comparativa

Realizamos uma análise comparativa entre os simuladores de robótica educacional apresentados na Seção 3.1. Os itens levados em consideração foram: possibilidade de montar um robô virtual no simulador, características do robô virtual, como tipos de bases e a presença de sensores e de atuadores, tipo de ambiente de programação (gráfico ou textual), possibilidade da utilização de múltiplos robôs, permissão para modificar o ambiente virtual, e possibilidade de enviar o programa desenvolvido para um robô real. Os atuadores foram separados em duas categorias: atuadores que interagem com objetos e outros atuadores.

Foram retirados desta análise comparativa os softwares *Blender* e *AutoDesk Inventor*, já que eles não são caracterizados como simuladores. Foram citados na Seção 3.1.1 apenas por permitirem a montagem de robôs virtuais.

Podemos observar, através da Tabela 3.1, que o simulador S-Educ, assim como quatro dos simuladores analisados, permite montar um robô virtual, inserindo nele atuadores, sensores e peças diversas. Os simuladores S-Educ, *RoboMind*, *KickRobot* e o Simulador

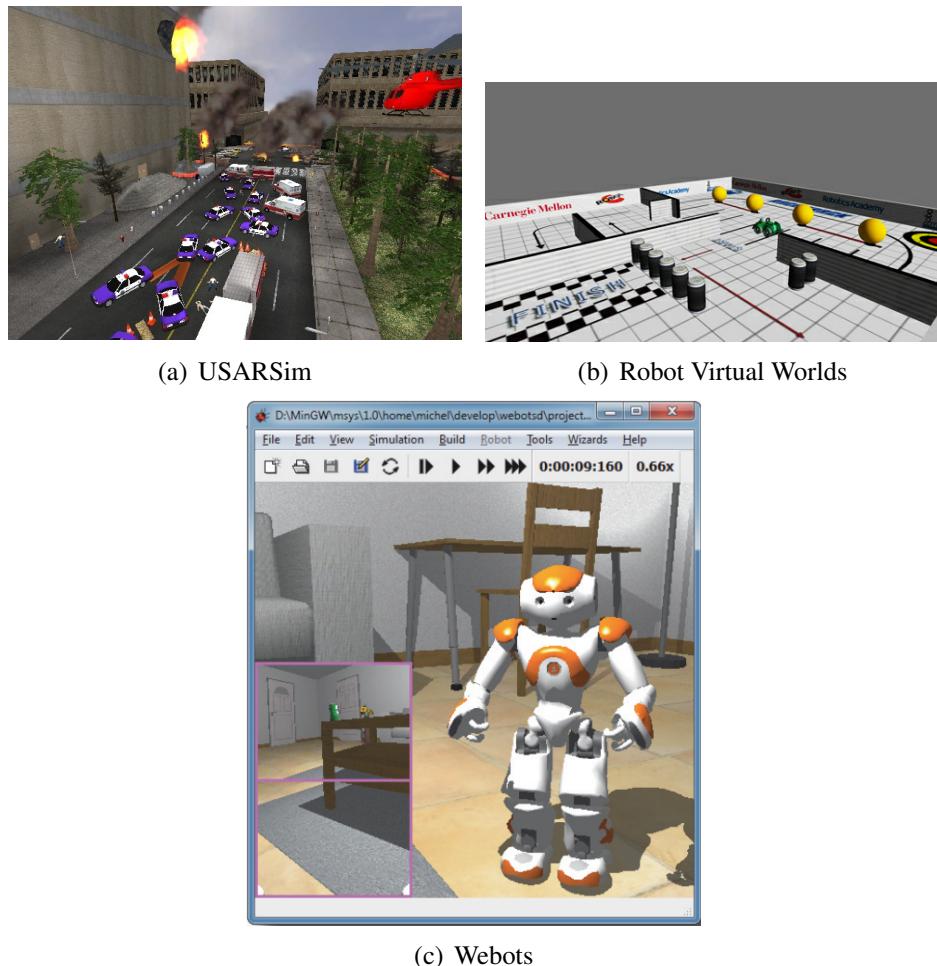


Figura 3.2: Simuladores para montagem e programação de robôs.

RoboEduc não permitem a utilização de robôs com diferentes tipos de bases. Destes, apenas o *KickRobot* pode utilizar robôs com base fixa, enquanto que os outros possuem robôs com bases móveis do tipo carro, que utilizam rodas ou esteiras.

Todos os robôs permitem a utilização de sensores, exceto o *KickRobot*. Com relação aos atuadores, apenas o S-Educ e o Simulador RoboEduc não permitem a utilização de atuadores que interagem com objetos, ou seja, atuadores capazes de pegar, empurrar ou mover objetos do ambiente virtual. Embora não possua este tipo de atuador, o S-Educ possui outros tipos de atuadores, como os atuadores do tipo carimbo e caneta, capazes de marcar o ambiente virtual com diferentes cores. Outros dos simuladores apresentados também permitem a utilização de outros tipos de atuadores.

Dos simuladores apresentados, apenas quatro deles permitem que o programa desenvolvido e testado no robô virtual seja enviado para o robô real, entre eles o S-Educ. Permitir o teste também em robôs reais faz com que o usuário possa comparar o funcionamento

	<i>LDD</i>	<i>RoboMind</i>	<i>KickRobot</i>	Simulador RoboEduc	<i>USARSim</i>	<i>Robot Virtual Worlds</i>	<i>Webots</i>	S-Educ
Montar o robô	•				•	•	•	•
Robô com diferentes tipos de bases	•				•	•	•	
Robô com sensores	•	•		•	•	•	•	•
Robô com atuadores • Atuadores que interagem com objetos • Outros atuadores	•	•	•		•	•	•	
•	•				•	•	•	•
Programar o robô • Programação gráfica • Programação textual				•				•
•	•	•		•	•	•	•	•
Enviar programa para robô real				•	•		•	•
Múltiplos robôs	•				•		•	•
Visualização • Bidimensional • Tridimensional		•	•	•		•		•
•	•				•	•	•	•
Modificar ambiente virtual	•	•		•	•	•	•	•

Tabela 3.1: Tabela comparativa entre os Simuladores de Robótica Educacional.

do robô real e virtual.

A programação no simulador S-Educ pode ser visual ou textual, assim como no simulador 2D prévio do RoboEduc. Os outros simuladores permitem a programação do robô apenas em ambiente textual. Além disso, o S-Educ permite a utilização de múltiplos robôs, o que permite o desenvolvimento de atividades mais complexas. Os simuladores LDD, *USARSim* e *Webots* também permitem a utilização de múltiplos robôs.

Com relação ao tipo de visualização, apenas o S-Educ e o *Robot Virtual Worlds* permitem que o usuário visualize o ambiente tanto bidimensional quanto tridimensionalmente. Esta função facilita a realização de certas atividades, principalmente se forem ser utilizadas por crianças. Por fim, apenas o simulador *KickRobot* não permite a modificação do ambiente virtual, ou seja, não permite adicionar, remover ou mover objetos virtuais, o que o torna mais limitado com relação a criação de aulas de robótica educacional.

Analizando a Tabela 3.1, pode-se verificar que os simuladores *USARSim* e *Webots* preenchem quase todos os itens da tabela. Apesar de possuírem todas essas funcionalidades, estes simuladores são muito complexos para serem utilizados por alunos de faixa etária mais baixa, como, por exemplo, alunos de ensino fundamental. Nesse ponto, o simulador S-Educ se destaca, por ser um simulador simples de ser utilizado por crianças e adoles-

centes, e mesmo assim possuir funcionalidades que permitem que ele possa ser utilizado em aulas de robótica educacional.

Capítulo 4

Desenvolvendo o Simulador S-Educ

Neste capítulo, será apresentado o simulador para aplicações em robótica educacional, chamado S-Educ, bem como as suas funções e as principais características dos robôs e do ambiente virtual. Este simulador é utilizado como um *plugin* do *software* educacional R-Educ, nosso *software* para programação de robôs, usado na metodologia robótica educacional RoboEduc [Silva 2009].

A proposta inicial deste trabalho é desenvolver um simulador para ser utilizado em aulas de robótica educacional, mas, sabendo dos benefícios da utilização de *kits* de robótica pelos alunos, nós buscamos uma forma de que o programa desenvolvido e testado no robô virtual pudesse ser testado também em robôs reais. Por este motivo, construímos um robô virtual que simulasse um modelo de um robô real.

O *software* R-Educ pode ser utilizado para programar robôs do tipo *Lego RCX*, *Lego NXT* ou ainda usando a plataforma H-Educ [Sá 2011]. O *kit* *Lego RCX* é um modelo obsoleto, que já saiu do mercado, e o *kit* H-Educ é um modelo desenvolvido por pesquisadores do Laboratório Natalnet, que necessita de mais testes de usabilidade, não tendo se tornado, ainda, um modelo padrão que possa ser adotado de forma ampla. Por este motivo, os robôs virtuais do simulador S-Educ simulam robôs do tipo *Lego Mindstorms NXT*, possuindo características semelhantes às dos robôs construídos com este *kit*. Os robôs virtuais possuem a mesma aparência dos robôs *Lego*, os mesmos sensores disponíveis no *kit* *Lego NXT*, além da semelhança com relação à velocidade dos robôs, dependendo de como o robô foi montado. Convém ressaltar que a construção de modelos virtuais para outras plataformas é uma tarefa direta, que envolve apenas trabalho braçal de algum programador/modelador.

4.1 Software Educacional R-Educ

O *software R-Educ* é um *software* desenvolvido para ser aplicado em aulas de robótica educacional, que permite o controle e a programação de vários tipos diferentes de robôs. O R-Educ pode ser utilizado com os *kits Lego RCX*, *Lego NXT* e H-Educ. A programação do robô pode ser feita utilizando um dos cinco níveis de programação (abstração) existentes no R-Educ, que incluem programação visual ou gráfica e textual.

A linguagem de programação utilizada, também denominada de R-Educ, é uma linguagem em português estruturada, simples, que pode ser adotada para alunos a partir dos 8 anos [Barros 2011, Sá 2011]. A linguagem R-Educ possui comandos que permitem a movimentação de motores e a aquisição de dados dos sensores do robô, comandos de seleção e repetição, e outros comandos, responsáveis por imprimir textos ou acender luzes no robô. A Tabela 4.1 mostra exemplos de comandos da linguagem R-Educ.

Os comandos escritos em linguagem R-Educ são compilados no ambiente R-Educ. O programa é então lido e interpretado pelo simulador S-Educ, que gera comandos que podem ser executados pelo robô virtual.

Comandos	Exemplos
Motores	<i>Frente, Direita, Carimbar</i>
Sensores	<i>CorBaixoMeio, ToqueFrente, UltraEsquerda</i>
Repetição e Seleção	<i>Enquanto, Repita, Para, Se</i>
Outros	<i>Texto, Luz</i>

Tabela 4.1: Exemplos de comandos em linguagem R-Educ.

4.2 Modelagem do robô virtual

Para se criar um modelo para simulação virtual, é necessária uma modelagem simplificada da realidade, desconsiderando o que não for necessário e/ou relevante para a simulação dos componentes reais do robô. O modelo deve conter os elementos e as características mais importantes para o funcionamento do robô virtual. Os principais pontos a serem modelados são os movimentos do robô, seus sensores e atuadores.

4.2.1 Movimentos do robô

Para a modelagem da movimentação do robô virtual levamos em consideração as características dos movimentos dos robôs reais. Considerando robôs do tipo carro, existem

características que modificam dependendo do tipo da base do robô. Por exemplo, robôs com bases com quatro rodas andam muito bem na reta, mas não tão bem nas curvas.

Modelamos quatro tipos diferentes de bases: base com quatro rodas, com quatro rodas e engrenagens, com duas rodas e uma roda boba, e com esteira. A Figura 4.1 mostra as bases que foram modeladas para o simulador S-Educ. Realizamos uma série de testes para definir a velocidade de cada uma destas bases, tanto na reta quanto na curva. A Tabela 4.2 mostra as velocidades de cada uma destas bases nas duas diferentes situações: reta e curva. Durante os testes, foram utilizados robôs com pilhas com carga média, e força máxima do motor.

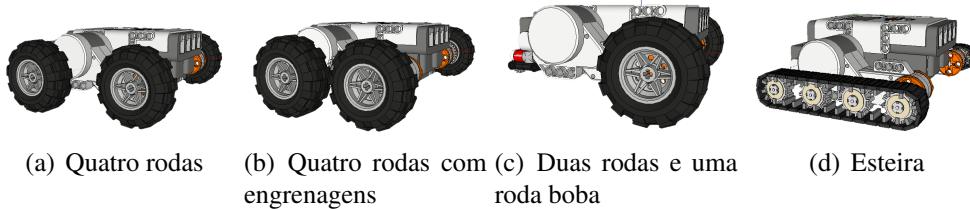


Figura 4.1: Bases modeladas para o simulador S-Educ.

Tipo de Base	Velocidade linear média	Velocidade angular média
Quatro rodas	24.03 cm/seg	50.42 °/seg
Quatro rodas e engrenagens	22.32 cm/seg	82.94 °/seg
Duas rodas e uma roda boba	25.38 cm/seg	152.54 °/seg
Esteira	13.40 cm/seg	60.20 °/seg

Tabela 4.2: Velocidades dos diferentes tipos de bases.

As ações executadas pelo robô podem ser descritas em segundos ou em rotações do motor (Algoritmo 1). No primeiro caso, é especificada a duração, em segundos, da execução da ação, e no segundo caso é especificada a quantidade de rotações do motor. Para contar a duração das ações descritas em segundos, é usado um temporizador, que indica quando o tempo de execução daquela ação terminou. No caso das ações descritas em rotações, foi feita uma análise do tempo de duração de uma rotação nas diferentes bases de robôs. Dessa forma, também é utilizado um temporizador para contar quando a quantidade de rotações do robô terminou.

No robô real, o movimento executado pelo robô quando a ação é descrita em segundos pode variar dependendo da bateria do robô, ou seja, um robô com baterias totalmente

carregadas anda mais rápido que um robô com as baterias fracas, se for considerado o mesmo intervalo de tempo. Esta diferença não foi levada em conta para a modelagem do robô virtual, pois, neste caso, não há perda de força da bateria. Por esse motivo, é válido converter as rotações do motor em segundos, coisa que não seria possível no caso de robôs reais.

Algorithm 1 Exemplos de ações em segundos e em rotações na linguagem Educ.

```
tarefa movendo_robo
inicio
    Frente 2 segundos
    Direita 3.5 rotacoes
fim
```

Durante a execução dos movimentos do robô, os motores são suscetíveis a erros, proveniente do tempo de resposta do motor, ou seja, o tempo que o motor leva para parar de girar ao fim de um comando. Este erro resulta em orientações e posições diferentes das esperadas. Por exemplo, quando se comanda um giro, muito provavelmente não se atingirá o ângulo previsto.

Para simular tal comportamento utilizamos uma adição de erro randômica. Em cada tempo de duração de uma ação é adicionado um valor randômico, dentro de uma faixa de valores pré-definida, que descreve o tempo que leva para o motor parar de girar. Isso gera pequenas falhas de execução que são aceitáveis, visto que simulam falhas existentes no robô real.

Outras questões com relação ao movimento do robô, como força, gravidade e colisões, foram utilizadas na modelagem do robô virtual, e são representadas pelo *JMonkey Engine*. As questões referentes a equilíbrio e fricção não foram utilizadas para a representação do robô virtual.

4.2.2 Sensores

Os robôs do simulador S-Educ simulam robôs do tipo Lego NXT, e por esse motivo foram modelados os sensores existentes neste *kit* de robótica: sensor de toque, sensor de cor e sensor ultrassônico.

O sensor de toque é capaz de identificar a colisão com algum objeto, através do pressionamento de um botão existente no sensor. Ele foi representado por uma esfera que pode ser localizada na frente e/ou atrás do robô. O pressionamento do sensor de toque virtual

acontece quando há uma colisão entre ele (a esfera que o representa) e algum objeto do ambiente virtual. O sensor de toque virtual está representado na Figura 4.2(a).

O sensor de cor do *kit* Lego NXT é um sensor que permite identificar 6 diferentes cores: preto, azul, verde, amarelo, vermelho ou branco. A leitura do sensor só é correta quando ele está localizado a no máximo 1 cm do objeto. Caso contrário, o sensor identificará a cor como preta. O sensor de cor pode ser utilizado também como lâmpada, emitindo uma luz pontual azul, vermelha ou verde.

No modelo virtual, o sensor de cor é representado por uma esfera, localizada na frente do robô em quatro posições diferentes: no centro do robô e virado para frente, ou na parte de baixo do robô e virado para baixo, localizado no canto direito, no canto esquerdo ou no centro do robô. Como para a identificação da cor do objeto, o sensor deve estar bem próximo dele, definimos que a cor só será identificada quando a esfera que representa o sensor colidisse com o objeto. No robô virtual, podem ser inseridos, também, dois sensores de cor com a função de serem lâmpadas, virados para cima. A Figura 4.2(b) mostra o sensor de cor virtual.

No robô real, o sensor de cor está sujeito a erros de decisão, influenciados pela posição do sensor, iluminação do ambiente, brilho do objeto, entre outros fatores. Na modelagem do robô virtual, não foram levados em consideração estes fatores. No entanto, foi levada em consideração a utilização de objetos de cores diversas, podendo ser encontradas cores que não sejam uma das seis identificadas pelo sensor de cor. Neste caso, é feita uma análise do valor RGB encontrado, e ele é convertido em um dos valores identificados pelo sensor. O Algoritmo 2 mostra como essa conversão é realizada.

Por fim, foi modelado o sensor ultrassônico. O sensor ultrassônico é um sensor capaz de detectar a distância do robô a algum objeto. Este sensor produz um sonar do tipo cone, com um ângulo de 30° , e pode detectar objetos a uma distância de até 255 cm. Para modelar o sensor ultrassônico, o mesmo foi representado por um cone com um ângulo de 30° , que pode ser localizado em quatro diferentes posições: na frente, atrás, do lado direito ou do lado esquerdo do robô virtual. O tamanho do cone é determinado pela distância que se deseja testar. Para determinar se há algum objeto a uma determinada distância, é testado se houve alguma colisão entre este cone e os objetos do ambiente virtual. A Figura 4.2(c) mostra o sensor ultrassônico virtual.

A Figura 4.3 mostra exemplos de robôs virtuais com sensores de toque, cor e ultrassônico. Nas Figuras 4.3(a) e 4.3(b), as esferas vermelhas identificam os pontos de colisão dos sensores de toque e cor, enquanto que na Figura 4.3(c), o cone vermelho mostra os pontos de colisão do sensor ultrassônico.

Algorithm 2 Conversão de cores.

```

ColorRGBA newColor = null;
float r = color.getRed();
float g = color.getGreen();
float b = color.getBlue();

if (r > 0.5) { r = 1; }
else { r = 0; }
if (g > 0.5) { g = 1; }
else { g = 0; }
if (b > 0.5) { b = 1; }
else { b = 0; }

if (r == 0 && g == 1 && b == 1) { g = 0; } // Cor: azul
if (r == 1 && g == 0 && b == 1) { b = 0; } // Cor: vermelha

newColor.set(r, g, b, color.getAlpha());

```



(a) Sensor de Toque (b) Sensor de Cor (c) Sensor ultrassônico

Figura 4.2: Sensores virtuais.



(a) Sensores de toque (b) Sensores de cor (c) Sensor ultrasônico

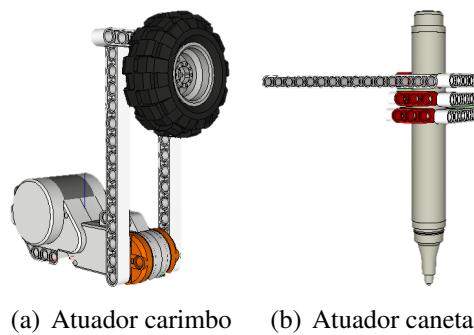
Figura 4.3: Sensores dos robôs virtuais.

4.2.3 Atuadores

Atuadores são partes do robô capazes de exercer movimento, através da movimentação de motores, modificando o ambiente no qual o robô está inserido. O simulador S-Educ possui, nesta versão, dois tipos de atuadores: um atuador carimbo e um atuador

caneta.

O atuador carimbo (Figura 4.4(a)) é um atuador no qual pode ser acoplado um algodão com tinta, que é capaz de marcar o ambiente de diversas cores através do movimento do motor. O atuador caneta (Figura 4.4(b)), embora não possua motores capazes de realizar movimentos, foi classificado como atuador por modificar o ambiente no qual está inserido. Neste tipo de atuador, é inserida uma caneta no robô e, dessa forma, o robô é capaz de marcar por onde andou.



(a) Atuador carimbo (b) Atuador caneta

Figura 4.4: Tipos de atuadores do simulador S-Educ.

4.2.4 Detecção de Colisões

Simular colisões é uma característica desejável em qualquer simulador de robô móvel, uma vez que permite simular situação real em que o robô se choca com objetos. Na plataforma proposta, os testes de colisão são realizados em duas situações: durante a movimentação normal do robô e durante os testes dos sensores. No primeiro caso, testamos se o robô está colidindo com objetos, paredes ou com outros robôs. Nesses casos, o robô deve parar de se mover, em caso de colisão frontal, ou mover em uma direção indicada pela força resultante da colisão. Em um segundo caso, o teste de colisão é usado para testar se os sensores estão detectando algum objeto. Por exemplo, testar se o sensor de toque deve ser acionado, se o sensor de cor detectou alguma cor, e se o sensor ultrassônico detectou a presença de algum objeto.

Testar colisões com objetos complexos, como os robôs, exige uma série de testes. Para simplificar esses testes, são utilizados *bounding volumes* (BV). Os BV são estruturas de dados simples, muito utilizadas em aplicações computacionais [Foley et al. 1996], que funcionam como uma envoltória simples que contém objetos mais complexos em seu interior. Para a escolha do *bounding volume* que envolveria o robô, analisamos os BVs do tipo esfera, cilindro e fecho convexo. Não utilizamos BV com faces lisas, como o cilindro, pois esses tipos de estruturas não resultam em uma boa movimentação do objeto.

Não utilizamos BV do tipo fecho convexo, embora eles representem o menor volume contendo o objeto, pois esta estrutura é mais complexa, e por esse motivo não resulta em resultados satisfatórios com relação à movimentação do robô. Enfim, escolhemos o BV do tipo esfera, pois este tipo resultou nos melhores resultados com relação à movimentação e encaixe do robô. A Figura 4.5 mostra o *bounding volume* de um dos robôs virtuais. Para os objetos virtuais são utilizados BV do tipo fecho convexo.

Os sensores, como descrito na Seção 4.2.2, são representados por esferas ou cones, e permitem a identificação de toque, cor ou presença, a partir da detecção de colisão com objetos virtuais. Por já serem representados por estruturas geométricas simples, não são utilizados *bounding volumes* neste tipo de detecção de colisão.

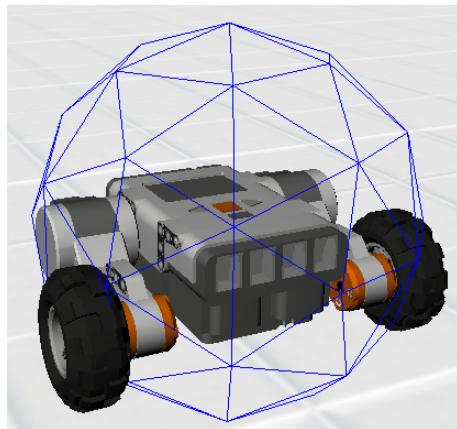


Figura 4.5: Modelo de um robô com *bounding volume*.

4.3 Interface Gráfica

As Figuras 4.6, 4.7 e 4.8 mostram as telas do simulador S-Educ. A Figura 4.6 mostra a tela de escolha entre *Aluno* ou *Professor*. O professor pode criar ou editar atividades, enquanto o aluno só pode utilizar as atividades criadas pelo professor. O ambiente de *Escolher atividade* (Figura 4.7(a)) permite visualizar os detalhes das atividades cadastradas.

A Figura 4.7(b) mostra a tela dos *Detalhes da atividade*, na qual são mostrados os detalhes da atividade selecionada, que podem ou não ser editados. Nesta tela também é possível inserir os detalhes de uma nova atividade que está sendo criada. Os detalhes da atividade são os seguintes:

- Nome
- Descrição

- Observações
- Configurações gerais
 - Modificar atividade
 - Adicionar elementos virtuais
 - Modificar ambiente virtual
 - Modificar câmera
 - Utilizar mais de um robô
- Imagens
- Vídeos

A Figura 4.8(a) mostra a tela do *Ambiente virtual*, que contém o ambiente de simulação, do lado direito da tela, com um conjunto de botões capazes de modificar a posição da câmera, e uma área do lado esquerdo da tela, que pode ser utilizada para duas funções (Figura 4.8(b)): montar o ambiente virtual e programar o robô. No primeiro caso, ela permite adicionar figuras, linhas, robôs e objetos no ambiente virtual, além de permitir editar os elementos virtuais já adicionados. No segundo caso, ela permite executar os programas para os robôs que foram adicionados no ambiente virtual, contendo informações sobre cada um dos programas que será executado, além de permitir exibir o texto impresso na tela dos robôs.

Por fim, a Figura 4.8(c) mostra a tela de *Montar robô*, na qual são adicionadas partes do robô para criar um modelo robótico que pode ser utilizado pelos usuários para executar uma determinada atividade.

4.4 Diagramas UML

Diagramas *UML* são notações gráficas que auxiliam na descrição e no projeto de sistemas de *software*. Neste trabalho foram analisados três diagramas *UML*: diagrama de classes, de casos de uso e de atividades.

4.4.1 Diagrama de Classes

Neste trabalho, adotamos o modelo de desenvolvimento orientado a objetos. Neste modelo, o diagrama de classes representa a estrutura do sistema, possuindo informações sobre métodos, atributos, nome das funções e como estas são integradas entre si. A Figura 4.9 representa o diagrama de classes do simulador S-Educ. Como pode-se ver, o sistema é constituído por seis classes principais. A classe *Interpreter* é responsável por interpretar



Figura 4.6: Tela de escolha entre aluno e professor

o programa em linguagem R-Educ, analisando os resultados obtidos pelos sensores, e enviando os comandos de movimento para o robô virtual. O principal atributo desta classe é *robo*, do tipo *ExecutarPrograma*. A classe *ExecutarPrograma* é responsável por executar os programas de cada um dos robôs interpretados pela classe *Interpreter*. As principais funções são *executarMovimento*, responsável por mover o robô, seguindo os comandos do programa, *adicionarCaminho*, que adiciona uma linha se o robô tiver um atuador do tipo caneta, *detectarColisao*, que detecta colisão entre dois robôs, criando um movimento de vai e volta, que simula o movimento real que acontece quando dois elementos se chocam, e *movimentoAtuador*, que faz o movimento do atuador carimbo, se o robô tiver este tipo de atuador.

A classe *AmbienteVirtual* é a classe que implementa o ambiente virtual. Ela possui atributos como *nosParaClicar*, que referencia todos os elementos que foram inseridos no ambiente virtual, e *noAtual*, que referencia o elemento que está atualmente selecionado. Dentre as funções, merecem destaque *simpleUpdate*, responsável pelo movimento do robô quando não há nenhum programa em execução, *detectarColisao* e *adicionarCaminho*, que possuem a mesma função que na classe *ExecutarPrograma*. O diagrama de classes (Figura 4.9) mostra que a classe *AmbienteVirtual* pode ter inúmeras instâncias da classe *Interpreter*, ou seja, vários programas podem ser interpretados e executados simultaneamente.

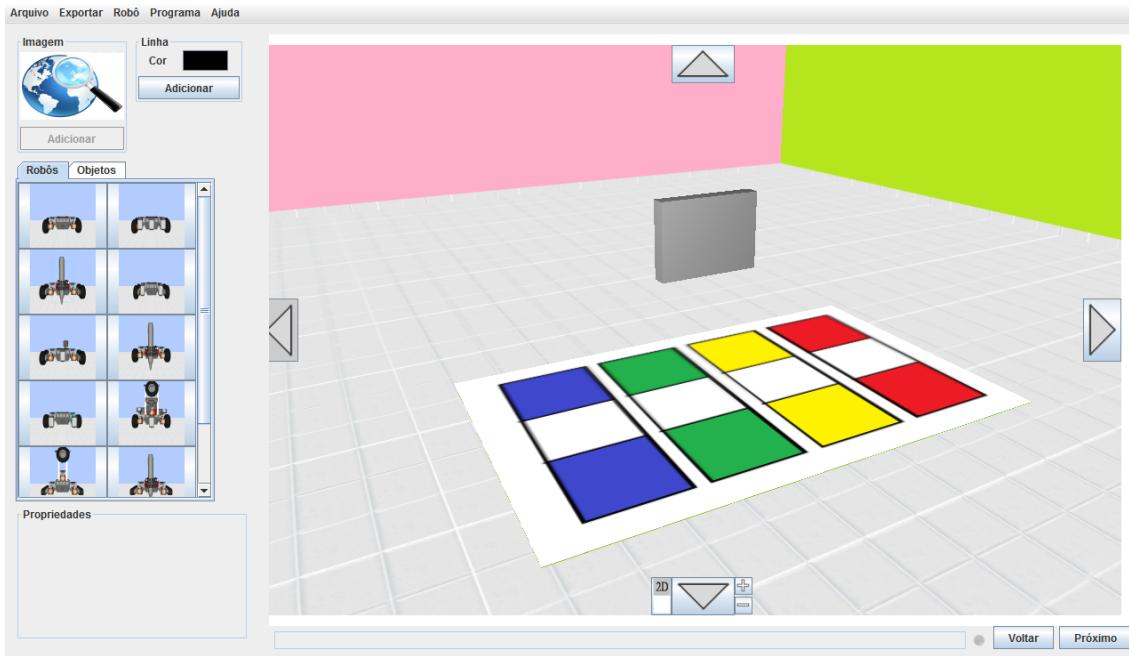
Ambas as classes *ExecutarPrograma* e *AmbienteVirtual* são responsáveis por mover o robô, e por isso ambas possuem uma função chamada *definirBase*, responsável por

(a) Escolher atividade

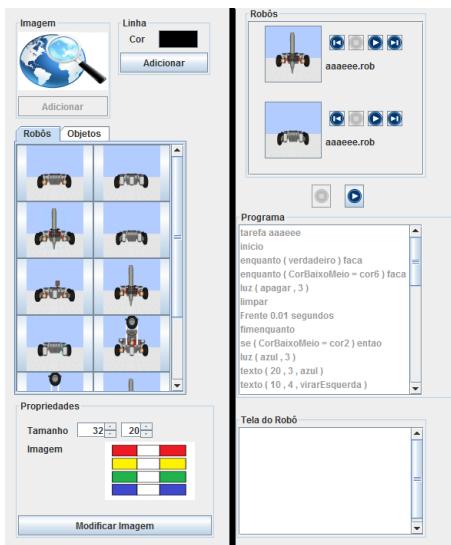
Modificar a atividade	Não
Adicionar elementos virtuais	Sim
Modificar o ambiente virtual	Sim
Modificar a câmera	Sim
Utilizar mais de um robô	Não

(b) Detalhes da atividade

Figura 4.7: Telas de atividades do simulador S-Educ.



(a) Ambiente virtual



(b) Montar x Programar



(c) Montar robô

Figura 4.8: Telas do ambiente virtual do simulador S-Educ.

definir a velocidade do robô dependendo do seu tipo de base, seguindo os dados obtidos em testes, e descritos na Seção 4.2.1.

A classe *Movimentos* possui as características do movimento que está sendo executado pelo robô, incluindo nome da ação, duração e tipo. Esta mesma classe é responsável por fazer a conversão da ação descrita em rotações por segundos. Por fim, a classe *Objetos* é responsável por inserir todos os elementos virtuais no ambiente.

As configurações do ambiente virtual, como luz e posição da câmera, são descritas pelas classes *AmbienteVirtual* e *CriarNovoRobo*. Esta última é instanciada apenas quando o usuário resolve criar um novo robô.

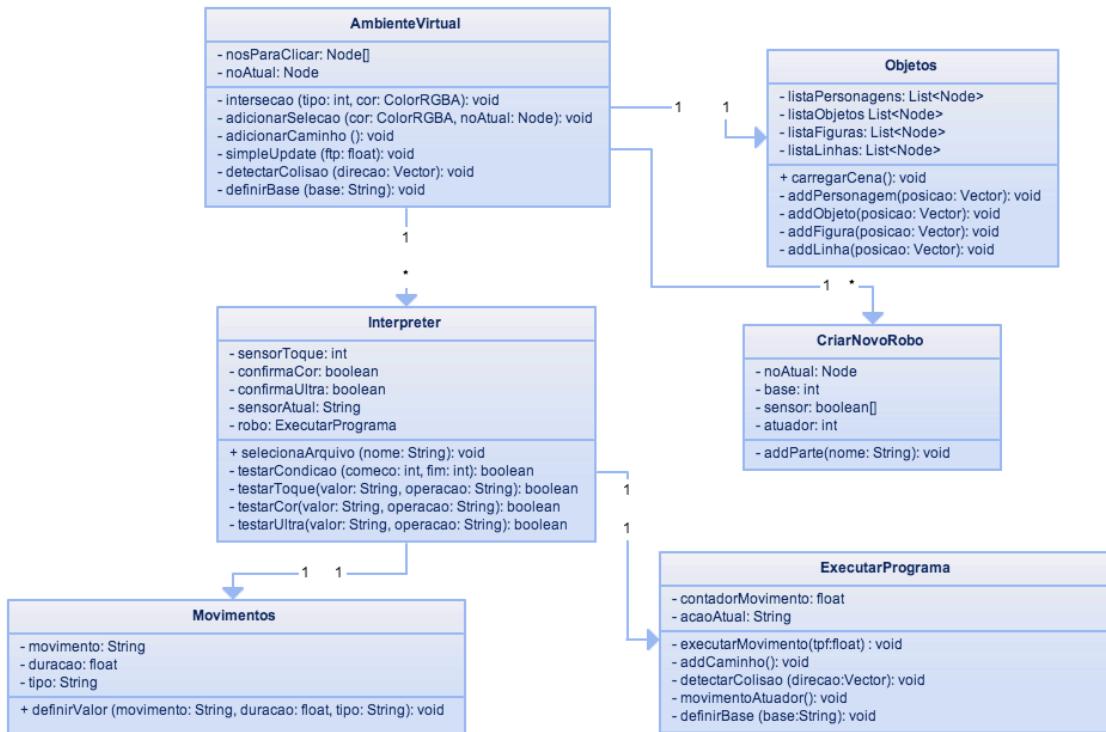


Figura 4.9: Diagrama de Classes do Simulador S-Educ.

4.4.2 Diagrama de Casos de Uso

No paradigma orientado a objetos, o diagrama de casos de uso é um modelo utilizado para a especificação de requisitos, descrevendo um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário. O diagrama de casos de uso do Simulador S-Educ pode ser visto na Figura 4.10.

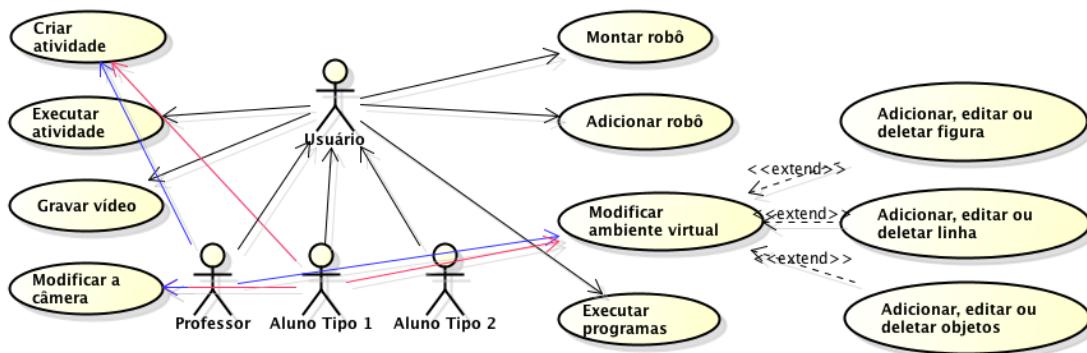


Figura 4.10: Diagrama de Casos de Uso do Simulador S-Educ.

O sistema tem basicamente três atores: professor, aluno tipo 1 e aluno tipo 2. Estes atores herdam seus atributos do ator usuário, herdando também todos os seus casos de uso. O professor é o usuário responsável por criar uma atividade que poderá ser executada pelos alunos, e possui total liberdade no sistema, tendo acesso a todas as funções do mesmo. O aluno é o usuário que utilizará o simulador para realizar uma atividade criada pelo professor. Existem dois tipos de alunos: aluno tipo 1 e aluno tipo 2. O aluno tipo 1 é aquele que está habilitado a realizar muitas das funções do sistema, inclusive modificar o ambiente virtual, com a adição de novos elementos ao mesmo. Já o aluno tipo 2 só está habilitado a utilizar o ambiente virtual definido pelo professor. O tipo de aluno que realizará a atividade é escolha do professor. A seguir, serão descritas todas as funcionalidades do sistema citadas no diagrama de casos de uso da Figura 4.10.

Criar atividade

O professor pode criar uma atividade para ser desenvolvida pelos alunos, composta por nome, descrição, observações, configurações gerais, imagens, vídeos e ambiente virtual.

O nome e a descrição da atividade contêm todos os detalhes necessários para que o aluno entenda o que é pedido nesta atividade. As observações contêm qualquer detalhe que o professor ache necessário ou interessante que o aluno tenha conhecimento.

As configurações gerais se referem a características do uso do simulador. Estas configurações caracterizarão o aluno que realizará tal atividade como aluno tipo 1 ou tipo 2. Existem cinco configurações gerais que devem ser definidas pelo professor:

- **Modificar atividade:** se o aluno pode modificar e salvar os dados da atividade

- **Adicionar elementos virtuais:** se o aluno pode adicionar elementos como objetos, figuras e linhas ao ambiente virtual
- **Modificar ambiente virtual:** se o aluno pode modificar, ou seja, mover ou deletar, os elementos virtuais que foram inseridos no ambiente virtual
- **Modificar câmera:** se o aluno pode modificar a câmera, de 3D para 2D e vice-versa
- **Adicionar mais de um robô:** se o aluno pode utilizar mais de um robô na realização da atividade

Além disso, o professor por escolher imagens e vídeos para auxiliarem o aluno na realização na atividade, ou no entendimento do tema abordado. Por fim, o professor deve definir um ambiente virtual para a realização da simulação. Caso não queira criar todo um ambiente, e prefira definir que o aluno irá montar um por si só, um ambiente virtual vazio será atribuído a tal atividade. A criação de uma atividade permite que o professor organize a aula previamente, facilitando o trabalho que ele terá durante sua realização.

Executar uma atividade

O aluno pode escolher uma das atividades, criada pelo professor, e a partir dela executar o que lhe foi pedido. O aluno tem a opção de salvar um vídeo ou uma foto do ambiente virtual, caso o professor tenha exigido o envio de tal mídia para a finalização da atividade.

Gravar vídeo

É possível gravar o ambiente virtual em um vídeo do tipo "avi", que pode ser posteriormente utilizado pelo aluno, para analisar o desenvolvimento da atividade e buscar possíveis erros, ou pelo professor, para fins de avaliação.

Modificar a câmera

Os usuários podem modificar a posição e/ou a orientação da câmera para auxiliar na visualização do ambiente virtual. Também é possível modificar a câmera de forma que o ambiente tridimensional pareça bidimensional. Ambientes bidimensionais são mais úteis em algumas atividades, e facilitam a visualização do ambiente virtual, principalmente quando estes ambientes estão sendo utilizados por crianças. Durante a criação da atividade, o professor habilita ou não a modificação do modo de visualização de 3D para 2D. A Figura 4.11 mostra o exemplo do mesmo ambiente virtual em dois modos de visualização diferentes: tridimensional e bidimensional.

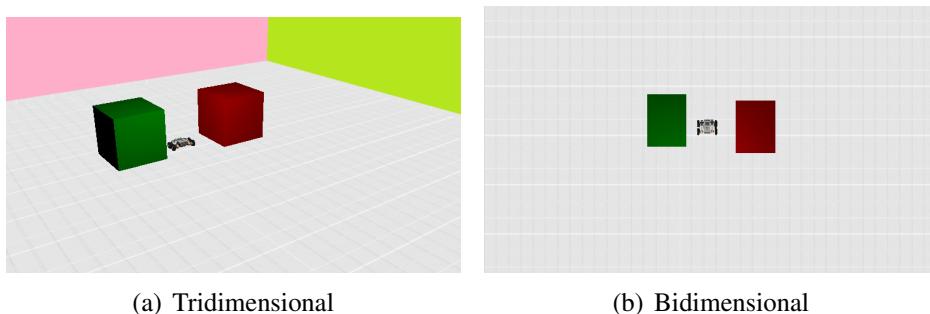


Figura 4.11: Visualização 3D e 2D do mesmo ambiente virtual.

Montar robô

Os usuários podem montar diferentes tipos de robôs virtuais para que possuam os requisitos necessários para realizar a atividade proposta. A montagem dos robôs virtuais é feita a partir da escolha de elementos de três blocos diferentes: bases, atuadores e sensores. O usuário pode fazer diferentes combinações, criando diversos tipos distintos de robôs, levando em consideração que as diferenças entre cada objeto dos blocos resultam em características distintas para o robô.

A escolha da base resultará em uma diferença na velocidade do robô. Por exemplo, robôs com 2 esteiras são mais lentos do que robôs com 4 rodas. A escolha dos atuadores vai depender do tipo de atividade que se deseja executar, sendo o usuário responsável por escolher, entre os atuadores disponíveis, qual será mais apropriado para o desenvolvimento da atividade proposta. Por fim, a escolha dos sensores depende do tipo de funcionalidade que será exigida do robô virtual. Por exemplo, caso seja necessário que o robô não toque em nenhum objeto, o usuário deverá inserir um sensor ultrassônico virado para frente.

As posições dos elementos do robô já são previamente definidas, ou seja, o usuário não pode adicionar uma parte do robô em qualquer local que desejar. Por exemplo, a Figura 4.12 mostra as possíveis posições do sensor ultrassônico no robô virtual, marcadas pelos paralelepípedos verdes. As peças que podem ser adicionadas nos robôs são mostradas nas Figuras 4.1 (bases), 4.4 (atuadores) e 4.2 (sensores).

Adicionar robôs

O usuário pode adicionar robôs ao ambiente virtual. Para adicionar um robô, deve-se montar o seu próprio robô virtual (Seção 4.4.2) ou escolher entre os modelos já montados, disponíveis ao usuário. Dependendo das permissões da atividade, o aluno pode ou não adicionar mais de um robô no ambiente virtual.

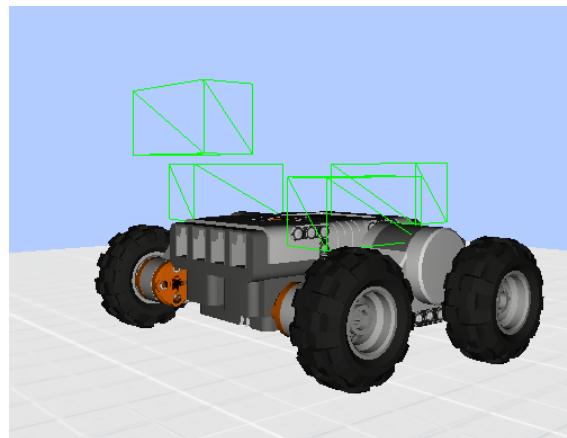


Figura 4.12: Possíveis localizações do sensor ultrasônico na montagem do robô virtual.

Modificar o ambiente virtual

O professor e o aluno tipo 1 tem a opção de alterar o ambiente virtual, inserindo, editando ou deletando elementos como objetos, linhas ou figuras.

Os objetos são elementos que podem ser inseridos no ambiente virtual para fazer com que o robô interaja mais com o ambiente, como por exemplo, na Figura 4.11(a), em que objetos foram inseridos para delimitar espaços nos quais o robô não pode passar. Quando adicionados, as dimensões dos objetos podem ser editadas. Os objetos podem ser identificados por sensores de toque, cor ou ultrassônico, podendo ser identificada sua presença ou sua cor.

As linhas são pedaços de retas de diversas cores que podem ser identificadas pelos sensores de cor do robô (apenas aqueles virados para baixo). A dimensão e a cor das linhas podem ser editadas posteriormente.

Por fim, também podem ser inseridas figuras no ambiente virtual, simulando cartolinhas, fotos, pinturas, etc. Depois de inseridas, as figuras podem ter seu tamanho e imagem modificados. Estes elementos não interagem com nenhum dos sensores presentes nos robôs virtuais. Eles apenas modificam o ambiente, podendo ajudar na criação de uma atividade por parte do professor.

Executar programas

Todos os usuários do simulador podem executar algum programa escrito em linguagem R-Educ. Para que o programa seja executado, deve haver pelo menos um robô inserido no ambiente virtual. O usuário escolhe um programa para cada um dos robôs virtuais, e a execução desses programas pode ser iniciada simultaneamente, sendo todos os progra-

mas executados ao mesmo tempo (claro, o paralelismo é aparente ao usuário). A Figura 4.8(b) mostra a área de execução de programas do simulador S-Educ.

4.4.3 Diagrama de Atividades

O diagrama de atividades representa os fluxos conduzidos por processamentos, mostrando o fluxo de controle de uma atividade para outra, e é modelado como uma sequência estruturada de ações, controladas por nós de decisão.

Como visto na Seção 4.4.2, o simulador S-Educ pode ser utilizado por um professor, responsável por cadastrar atividades no simulador, ou pelos alunos, que podem realizar as atividades cadastradas pelo professor. No diagrama de atividades da Figura 4.13, pode-se identificar a diferença entre os usuários professor, aluno tipo 1 e aluno tipo 2 durante a utilização do simulador S-Educ.

No diagrama de atividades (Figura 4.13) pode-se ver que o primeiro passo para a utilização do simulador é a execução de dois programas em paralelo: o simulador S-Educ e o *software* R-Educ.

Os passos da execução de uma atividade no simulador S-Educ são os seguintes: abrir os dois programas, montar ou editar o ambiente virtual (dependendo do tipo de usuário), montar um robô ou escolher outro modelo já montado, programar robô(s) no *software* R-Educ e testar o robô.

Durante a realização de testes, pode-se deparar com quatro situações distintas: teste com erro no ambiente virtual, teste com erro no robô virtual, teste com erro no programa, ou teste sem erros. No primeiro caso, deve-se retornar à área de montar ou editar o ambiente virtual. Alunos do tipo 2 não podem realizar este tipo de ação, e por isso, nesta situação, o usuário é levado a um estado de cancelamento, em que não é possível concluir a atividade. Outros usuários, ou seja, alunos tipo 1 ou professores, retornam ao ambiente virtual, e em seguida continuam a realizar os testes, pois estão em uma situação de testes, não sendo necessário escolher um novo robô e programar novamente o(s) robô(s).

No segundo caso, em que há teste com erro no robô virtual, o usuário retorna à área de escolha do robô, onde pode ser montado um novo robô ou escolhido outro robô já montado. Como a situação também é de testes, o usuário vai direto à realização de testes, sem ser necessário programar o(s) robô(s) novamente.

No terceiro caso, em que há teste com erro no programa, o usuário retorna ao *software* R-Educ e realiza modificações no programa previamente desenvolvido. No último caso, o teste realizado não apresenta erros, e com isso é atingido o final da programação com sucesso.

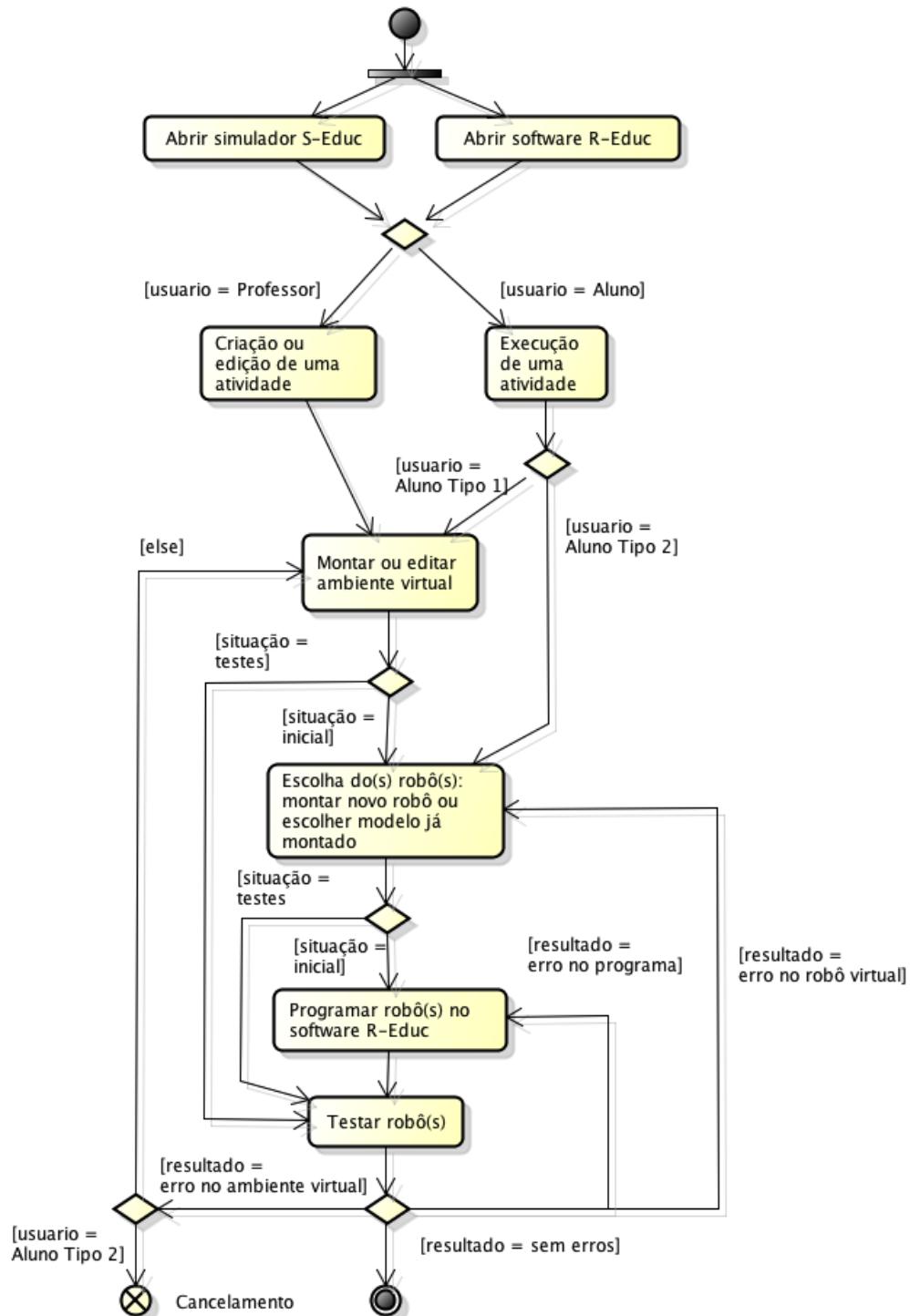


Figura 4.13: Diagrama de atividades do simulador S-Educ.

Capítulo 5

Experimentos e Resultados

O projeto e desenvolvimento do simulador S-Educ levou em consideração a proposta de ser utilizado por alunos de ensino fundamental em aulas de robótica educacional. Realizando uma análise a partir desta primitiva, percebemos que o simulador deveria possuir uma série de atributos capazes de tornar a aula de robótica educacional no ambiente virtual o mais real possível. Dentre os atributos, vale destacar a utilização de múltiplos robôs, a inserção de objetos para compor o ambiente, e a montagem do robô no ambiente virtual.

Neste capítulo apresentaremos cinco atividades que foram desenvolvidas para testar as funcionalidades do simulador, que envolvem a criação do ambiente de simulação e os testes de programação do robô neste ambiente. Estas atividades foram utilizadas nos testes realizados com o simulador S-Educ, nos quais foram testadas as funcionalidades do mesmo por três grupos diferentes de usuários: professores, alunos e adultos leigos em robótica.

5.1 Plataforma de testes

Desenvolvemos uma plataforma de testes para ser utilizada por todos os usuários do grupo de teste, que envolve a utilização de algumas das funcionalidades do simulador S-Educ, em atividades diferentes, dependendo da escolaridade e do nível de conhecimento em robótica e em programação de cada usuário.

O teste é dividido em três etapas: conhecimento da ferramenta, utilização da ferramenta e apresentação do questionário. Na primeira etapa o professor responsável pela aula mostra aos alunos como funciona a ferramenta, ou seja, que funcionalidades ela possui e como utilizá-las. Em seguida, os alunos realizam duas atividades utilizando o simulador. Aos alunos que não possuem conhecimento em programação com a linguagem R-Educ, é apresentado apenas o simulador S-Educ, ou seja, não é feita uma aula para ensiná-los a programar com o *software* R-Educ. Quando necessário, é cedido um pro-

grama base para ser utilizado no desenvolvimento da atividade. A terceira e última etapa envolve a apresentação de um questionário qualitativo/quantitativo no qual o usuário do simulador descreve a utilização do mesmo. Os Apêndices A, B e C mostram os três tipos diferentes de questionários aplicados.

5.2 Testes iniciais

Desenvolvemos cinco atividades para testar o simulador S-Educ, que envolvem a criação de atividades, a montagem de robôs, a utilização de sensores e atuadores, o controle direto do robô e/ou a utilização de programas para controlar robôs. Algumas destas funcionalidades não são utilizadas em todas as atividades, apenas a montagem de robôs, que é necessária em todos os casos. Estas atividades foram idealizadas inicialmente apenas para testar as implementações do simulador, mas posteriormente decidimos incluí-las nos testes.

A primeira atividade desenvolvida (Figura 5.1) testa o controle do robô pelo usuário, que deve percorrer um labirinto até chegar ao fim do mesmo. O usuário deve montar um robô que possua um atuador do tipo caneta, que marca o caminho percorrido por ele.

A segunda atividade (Figura 5.2) testa a utilização de múltiplos robôs com bases diferentes. O usuário deve adicionar robôs com bases diferentes alinhados, e executar o mesmo programa para todos os robôs, que anda pra frente um determinado tempo, e depois gira em torno do eixo também por um determinado tempo. Este programa é fornecido pronto aos usuários do simulador.

A terceira atividade desenvolvida (Figura 5.3) testa a inserção e edição de figuras e objetos no ambiente virtual. Nesta atividade, o usuário deve inserir um mapa no qual o robô deve percorrer o ambiente, quadrado a quadrado, movendo-se do quadrado verde até o vermelho, contornando obstáculos, quando necessário. Quando chegar ao quadrado vermelho, deve acender uma luz e escrever, na tela do robô, que chegou ao fim. O usuário deve montar um robô sem atuadores e com um sensor de cor virado pra cima. Para o desenvolvimento desta atividade é fornecido um programa em linguagem R-Educ que possui a base para a criação do programa capaz de realizar esta atividade, e o usuário deve apenas modificar o programa.

A quarta atividade (Figura 5.4) foi a atividade de seguir linha, na qual o usuário deve utilizar dois sensores de cor voltados para baixo e um sensor ultrassônico, voltado para frente, para que o robô se move seguindo uma linha no chão, e contornando obstáculos quando os encontrar. Este tipo de atividade é utilizado em competições de robótica no Brasil e no mundo, tais como a Olimpíada Brasileira de Robótica e a *RoboCup*.

Na quinta atividade (Figura 5.5), intitulada *Células-Tronco*, o robô deve se mover por um ambiente procurando tecidos danificados, marcados por linhas coloridas no chão. Quando encontrar as linhas, o robô deve marcar aquele ponto com um carimbo, andar pra trás até tocar um obstáculo, virar, e continuar a recuperar outros tecidos danificados do ambiente. O robô deve possuir um sensor de cor virado pra baixo, um sensor de toque pra trás, e um atuador do tipo carimbo. Nesta atividade também foi fornecido um programa previamente desenvolvido em linguagem R-Educ, que deve ser modificado pelo usuário.

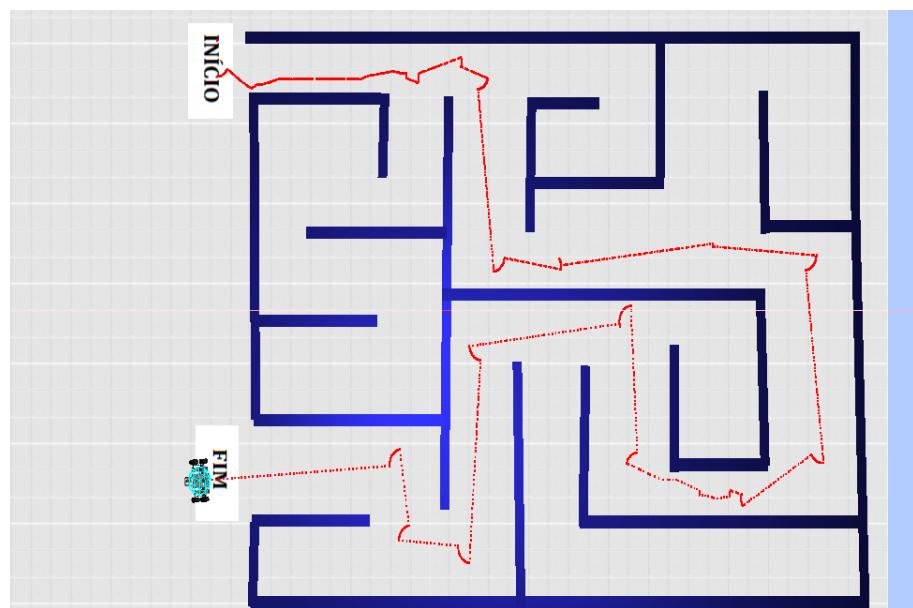


Figura 5.1: Atividade 1: Labirinto.

5.3 Grupos de testes

Como mencionado anteriormente, foram realizados testes com três grupos diferentes de usuários: professores, alunos e leigos em robótica. O primeiro grupo, composto por professores, envolve professores do ensino fundamental e professores de robótica. O grupo de alunos é constituído de alunos com e sem conhecimento em robótica e em programação. Por fim, o terceiro grupo, composto por leigos em robótica, envolve adultos que não possuem conhecimento de robótica ou programação de robôs. No total foram realizados testes com 60 usuários.

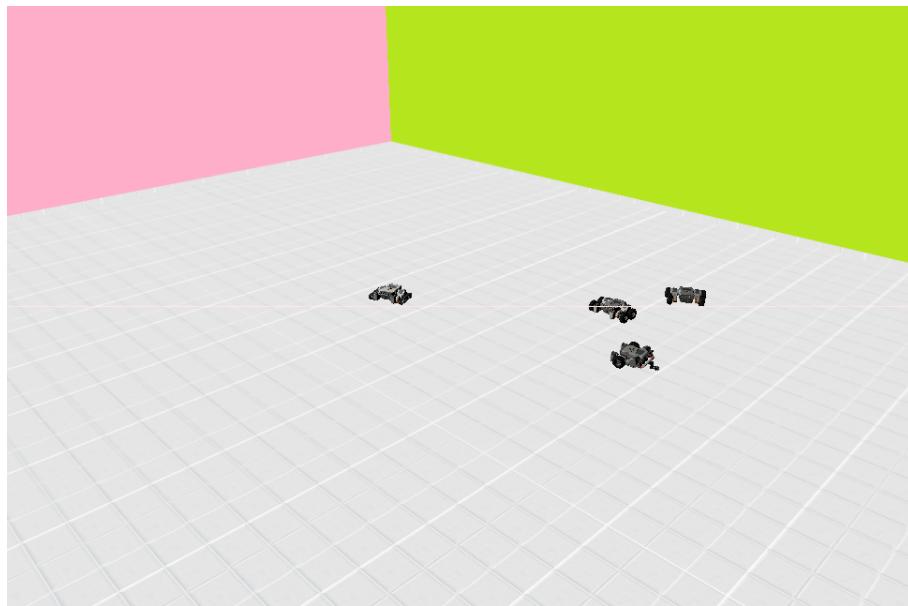


Figura 5.2: Atividade 2: Múltiplos Robôs.

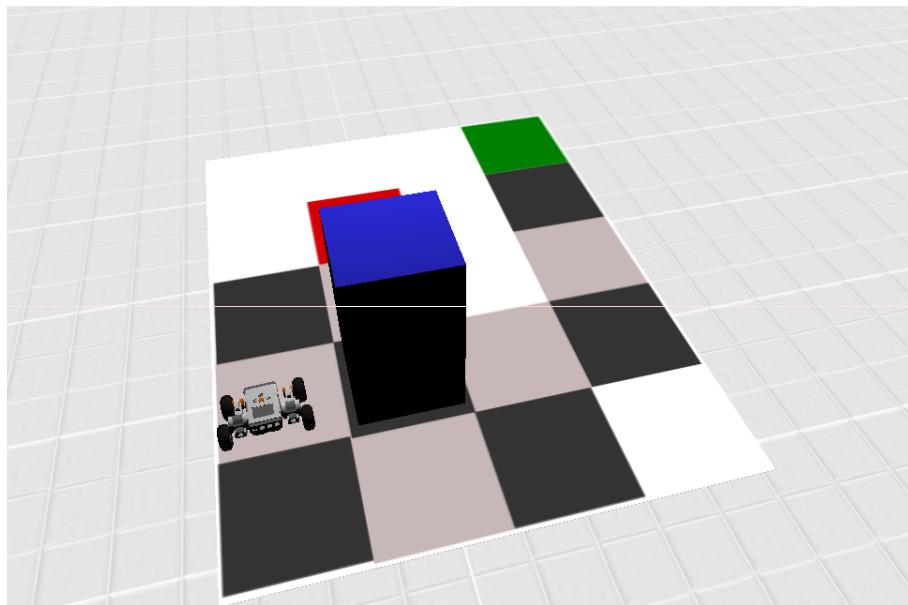


Figura 5.3: Atividade 3: Mapa.

5.3.1 Experimentos com professores

Realizamos testes com 11 professores, dos quais 2 são professores do ensino fundamental, com nível superior, e 9 professores de robótica, cursando nível superior nas áreas de tecnologia e de educação (no curso de pedagogia).

O teste foi separado em duas etapas: criação de atividade e realização de atividades.

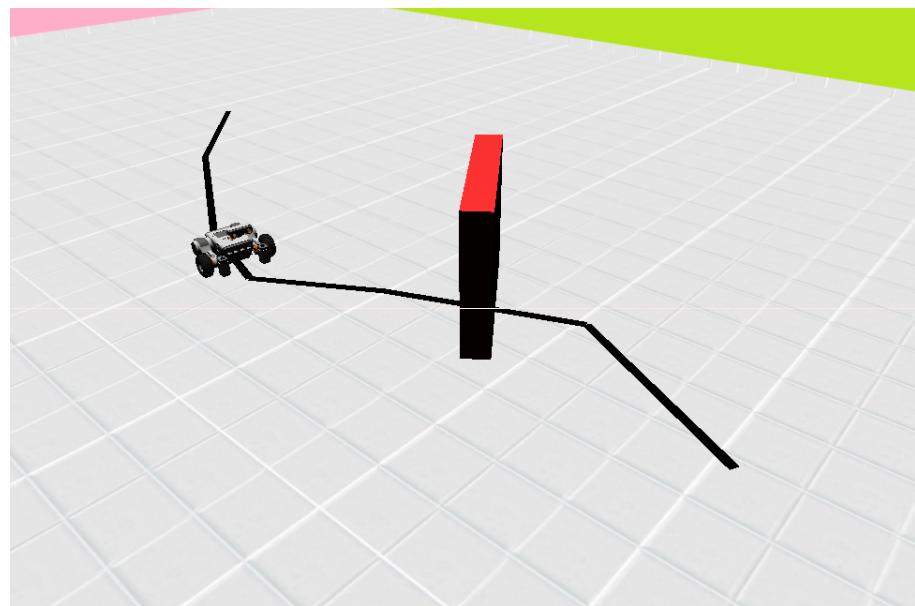


Figura 5.4: Atividade 4: Seguir Linha.

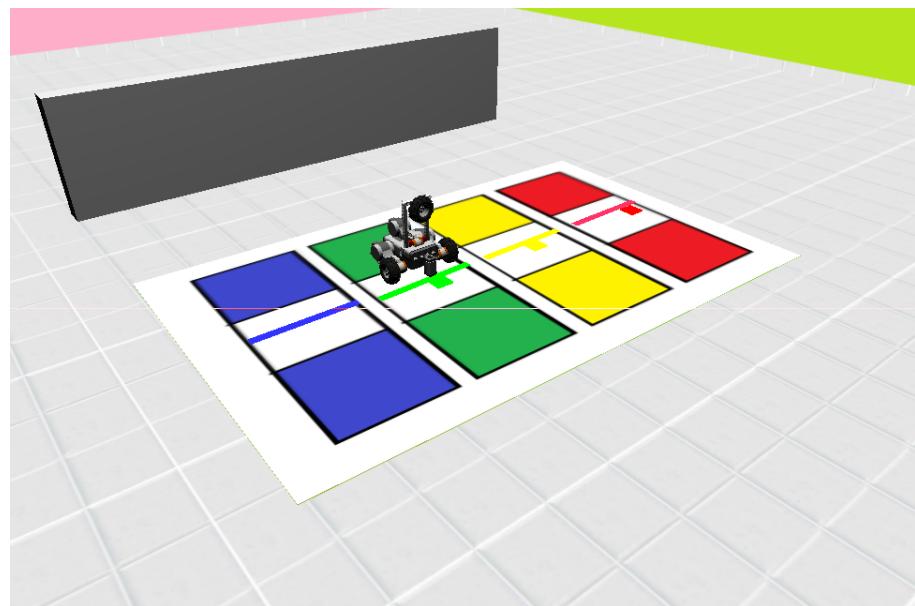


Figura 5.5: Atividade 5: Células-Tronco.

Na primeira etapa os professores devem criar uma atividade, intitulada Células-Tronco, com as características da atividade 5, descrita na Seção 5.2. Em seguida, na segunda etapa, os professores devem entrar no simulador como alunos e realizar duas dentre as quatro primeiras atividades descritas na Seção 5.2. Foi dado aos professores um questionário depois da realização do teste, no qual fora questionada, entre outros tópicos, a inserção deste tipo de ferramenta em sala de aula, a facilidade de utilização do simulador e a

veracidade da simulação.

A Figura 5.6 mostra alguns resultados desses testes realizados, em forma de gráficos. Os gráficos mostram que aproximadamente 65% dos professores consideram a ação de criar uma nova atividade fácil, 90% consideram a ação de montar robô fácil, e aproximadamente 45% classificam a utilização do ambiente virtual como fácil (Figura 5.6(a)). Na Figura 5.6(b) pode-se ver que aproximadamente 90% dos professores indicam que o simulador S-Educ simula muito bem o movimento do robô, levando em consideração que apenas os professores de robótica responderam a esta pergunta.

Por fim, os professores fazem uma avaliação da faixa etária para a qual o simulador S-Educ é mais adequado. A Figura 5.6(c) mostra um gráfico no qual pode-se ver que 9 dos 11 professores testados consideraram que o simulador é adequado para alunos das idades de 7 a 17 anos. Um dos professores indicou que o simulador é adequado para qualquer aluno apto a utilizar um computador.

5.3.2 Experimentos com alunos

Foram realizados testes com 37 alunos de 8 a 13 anos, dos quais 13 nunca tiveram aulas de robótica ou de programação, e 24 são alunos de cursos de robótica, que possuem também experiência com o R-Educ.

Dependendo da faixa etária e do nível de conhecimento em programação dos alunos, aplicamos atividades diferentes. Nos testes, realizamos com cada aluno, no total, duas das cinco atividades da Seção 5.2. Os alunos são separados em grupos, com dois ou três componentes para a realização da atividade.

Depois da finalização da atividade, aplicamos um questionário no qual os alunos respondem se acharam o simulador divertido, se o acharam fácil ou complexo, e se acharam que o simulador S-Educ simula bem o movimento de um robô, dentre outras perguntas.

A Figura 5.7 mostra resultados para os testes realizados com os alunos. Os alunos estão separados em dois grupos (grupos 1 e 2), sendo o grupo 1 composto por alunos que conhecem robótica e programação de robôs, e o grupo 2 composto por alunos que não possuem nenhum conhecimento nesta área. O primeiro gráfico (Figura 5.7(a)) mostra que, dentre todos os alunos, aproximadamente 60% consideram o simulador divertido. Este gráfico também mostra que a porcentagem de alunos que consideram o simulador muito divertido é maior em alunos do grupo 1 do que em alunos do grupo 2.

Os gráficos seguintes (Figura 5.7(b) e 5.7(c)) analisam a complexidade de montar o robô virtual e a de utilizar o ambiente virtual. Como se pode notar, a montagem do robô virtual é caracterizada como fácil por aproximadamente 97% dos alunos testados,

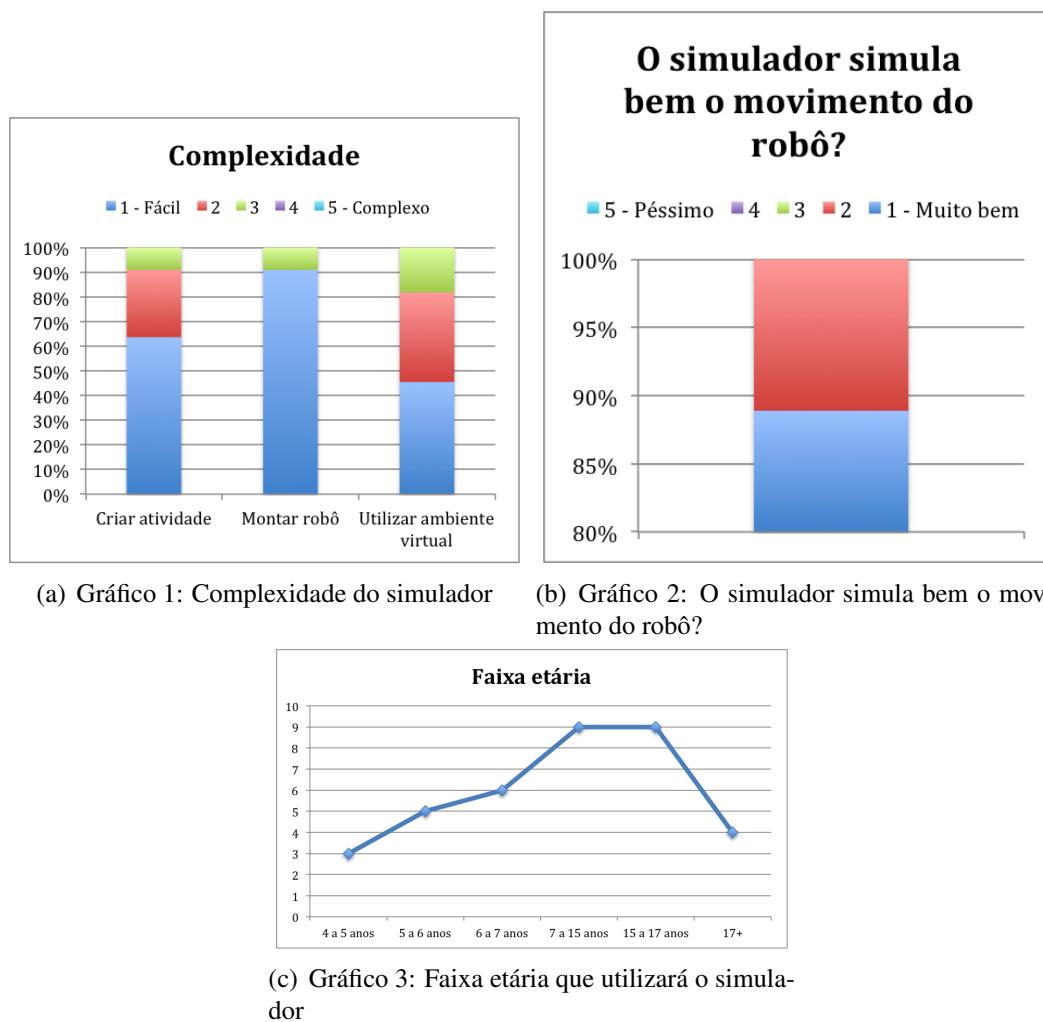


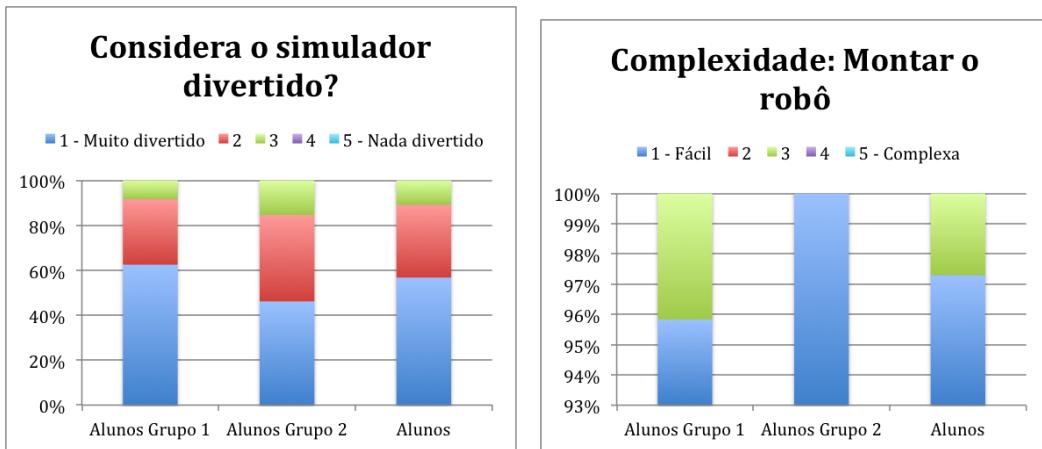
Figura 5.6: Gráficos das pesquisas feitas com o grupo de professores.

com exceção de um aluno do grupo 1, enquanto que a utilização do ambiente virtual é caracterizada como fácil apenas por 35% dos alunos, o que mostra uma diferença entre a complexidade da realização destas duas ações.

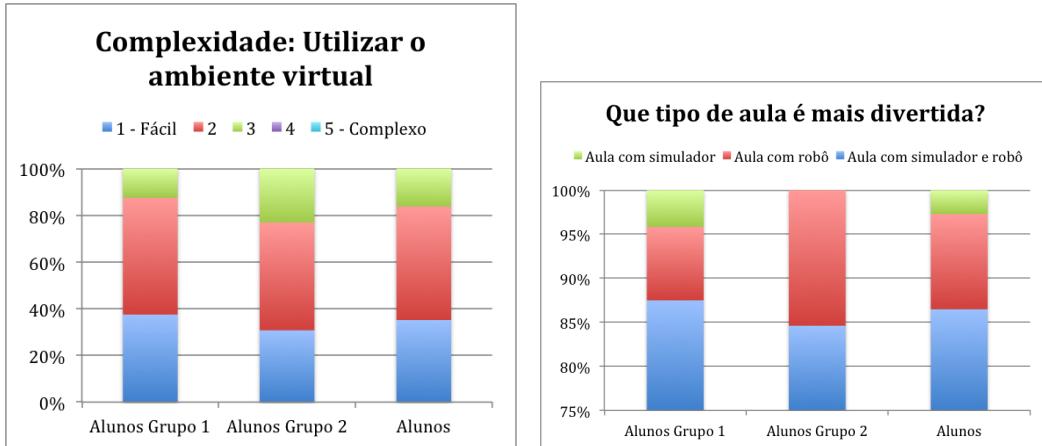
No grupo 1, composto por alunos que possuem conhecimento em robótica e em programação de robôs, 66% dos alunos acham que o simulador emula muito bem o movimento de um robô, enquanto que 34% classificaram a simulação do robô como 2 ou 3, em uma escala de 1 a 5, na qual 1 representa uma simulação muito boa.

Analisamos também que tipo de aula mais interessa os alunos (Figura 5.7(d)): aula com simulador, aula com robô ou aula com simulador e robô. No total, 86,49% dos alunos preferem aulas com simulador e robô, 10,81% preferem aulas apenas com o robô, enquanto que 2,7% dos alunos preferem aulas apenas com o simulador.

Com um subgrupo de testes, composto por 6 alunos, separados em duplas, que pos-



(a) Gráfico 1: Considera o simulador divertido? (b) Gráfico 2: Complexidade da montagem do robô virtual.



(c) Gráfico 3: Complexidade da utilização do ambiente virtual. (d) Gráfico 4: Tipo de aula que mais prefere?

Figura 5.7: Gráficos das pesquisas feitas com o grupo de alunos.

suem conhecimento em robótica e em programação, desenvolvemos uma atividade para analisar o tempo demandado para realizar a mesma atividade no ambiente virtual e no ambiente real, com robôs com bases diferentes. A atividade desenvolvida pelos alunos é a atividade 3, na qual eles devem andar em um mapa até chegar ao final do mesmo (Figura 5.3). O tempo demandado pelos alunos para realizar a atividade está mostrado na Tabela 5.1, na qual é possível ver que o tempo para a realização de testes no ambiente virtual é sempre menor do que o tempo para a realização do mesmo teste no ambiente real.

5.3.3 Experimentos com leigos

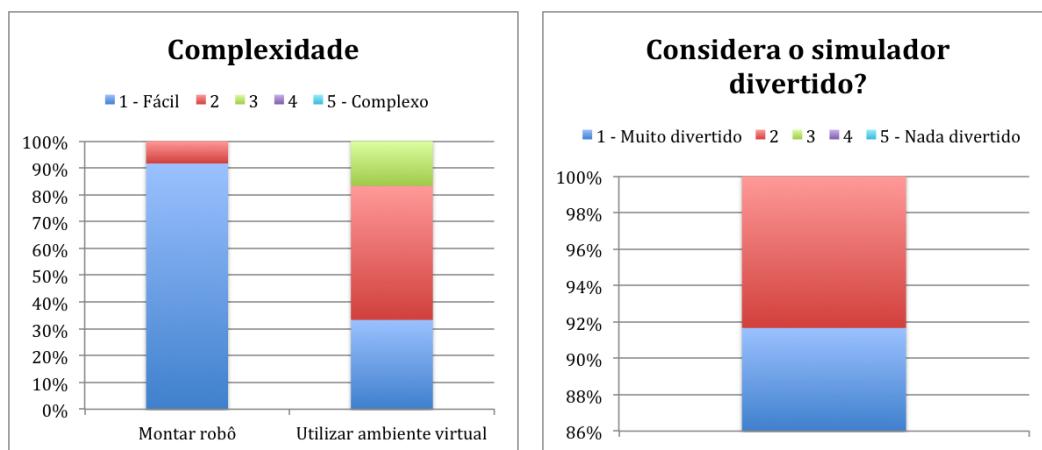
O terceiro grupo de pesquisa é composto por adultos leigos em robótica, para os quais são aplicadas duas das três atividades descritas na Seção 5.2. Depois de finalizada a ati-

Alunos	Tempo no simulador	Tempo no robô real
Grupo 1	5 minutos e 55 segundos	7 minutos e 10 segundos
Grupo 2	4 minutos e 20 segundos	5 minutos e 35 segundos
Grupo 3	5 minutos e 05 segundos	6 minutos e 40 segundos

Tabela 5.1: Tempo para a realização da atividade 3.

vidade, os usuários respondem um questionário no qual descrevem a complexidade de realizar tarefas como montar um robô e utilizar o ambiente virtual. Do grupo total, composto por 12 participantes, 90% classificam a montagem do robô como fácil, enquanto que 30% classificam a utilização do ambiente virtual como fácil, o que mostra que montagem do robô é uma atividade mais simples do que a utilização do ambiente virtual. Além disso, 92% dos usuários consideram o simulador S-Educ muito divertido. Resultados dessa pesquisa podem ser vistos nos gráficos mostrados na Figura 5.8.

Depois da finalização das atividades, em uma conversa informal, alguns dos participantes relataram pontos nos quais obtiveram dificuldades.



(a) Gráfico 1: Complexidade do simulador.

(b) Gráfico 2: Considera o simulador divertido?

Figura 5.8: Gráfico da pesquisa feita com o grupo de adultos leigos em robótica.

5.4 Análise dos Resultados

Realizamos testes com 60 participantes, separados em três diferentes grupos. Os participantes do grupo dos professores responderam perguntas na quais eles deveriam relatar críticas e sugestões para o simulador S-Educ. Muitas das críticas e sugestões relatadas envolvem a utilização do ambiente virtual. Os professores propuseram uma mudança na criação de linhas, devido à dificuldade na criação destes elementos virtuais, e uma me-

Ihora na nomenclatura dos elementos virtuais, que atualmente são chamados de linhas, figuras e objetos.

Com o grupo dos alunos e dos adultos leigos foi realizada uma conversa informal depois da realização da atividade, na qual eles ficaram a vontade para propor mudanças no simulador S-Educ. Muitos encontraram dificuldades na criação de linhas e na escolha do robô, pois depois que é feita a montagem de um robô, a figura dos robôs disponíveis para serem inseridos no ambiente virtual são bastante similares, e por isso não é muito fácil diferenciá-los. Além disso, muitos propuseram a utilização de apenas um *software*, pois acharam que utilizar um *software* para programar e outro para simular é muito trabalhoso. Eles propuseram também a utilização de outros tipos de atuadores, principalmente atuadores que permitam carregar objetos.

Capítulo 6

Conclusão

Almejando tornar a robótica educacional mais acessível às escolas brasileiras, planejamos e desenvolvemos um simulador robótico, chamado S-Educ, que pode ser utilizado em aulas de robótica educacional, em substituição ou em complementação ao uso de *kits* de robótica, tradicionalmente usados. Percebemos esta lacuna na metodologia em questão, que pode ser muito bem preenchida com o simulador, ao realizarmos pesquisa a respeito das ferramentas de simulação de robôs voltadas para a área educacional encontradas no mercado. Percebemos a necessidade de desenvolver um *software* que permita a criação de um ambiente virtual semelhante ao ambiente real, e que permita a utilização de todos os recursos robóticos encontrados em robôs reais, como sensores e motores.

A robótica educacional permite estabelecer um ambiente de aprendizagem rico, interdisciplinar, capaz de contribuir para o desenvolvimento da criatividade, do pensamento investigativo, do raciocínio lógico e da coordenação motora. Para a realização de uma aula de robótica seguindo o modelo tradicional, geralmente é necessário o uso de *kits* de robótica, que permitem a criação dos robôs. No entanto, a maioria dos *kits* possuem custo elevado, bem como possuem restrições com relação à qualidade de *software* e *hardware*.

O simulador S-Educ permite a criação de um robô virtual a partir da utilização de um conjunto de peças pré-definidas. O robô virtual pode possuir tipos diferentes de bases, atuadores e sensores. Ele pode ser diretamente controlado pelo usuário ou programado, a partir da descrição de comandos na linguagem R-Educ. Para a programação do robô, pode ser utilizado o *software* educacional R-Educ, que funciona em paralelo com o simulador S-Educ, mas de forma integrada e compatível.

Para o desenvolvimento de tal plataforma, utilizamos a linguagem *Java*, e o motor de jogo *JMonkey*, uma biblioteca que auxilia na construção de ambientes tridimensionais.

Visionamos várias contribuições deste trabalho à comunidade científica e também à sociedade. A utilização de simuladores por si só traz vários benefícios, como a diminuição do custo das aulas, do tempo demandado para os testes, da quebra de materiais, entre

outros. Assim, isso permite que o simulador S-Educ possa ser utilizado para implantar aulas de robótica em escolas brasileiras, fazendo com que crianças e adolescentes aprendam lógica de programação agregado a temas interdisciplinares de uma forma lúdica e atrativa. As publicações realizadas durante o desenvolvimento do simulador S-Educ mostram o mérito científico deste trabalho [Fernandes et al. 2012a, Fernandes et al. 2012b, Fernandes et al. 2012c].

A aula de robótica com simulador pode acontecer de duas formas diferentes: com e sem a utilização de *kits* de robótica. No primeiro caso, o simulador é utilizado apenas para realizar os testes iniciais, já que a realização de testes com o simulador é mais rápida do que a realização de testes com o robô real. Quando esses testes forem finalizados, os alunos devem utilizar um robô para realizar os testes finais. Para isso, é necessário realizar pequenas modificações no programa desenvolvido, referentes apenas à duração das ações executadas, já que a lógica de programação já foi testada. Nesse caso, vários grupos podem dividir o mesmo robô, já que o tempo de utilização do robô é reduzido, diminuindo também o custo total da aula de robótica. No segundo caso, o simulador é a única ferramenta utilizada na aula, e por isso todos os testes são realizados nele.

Para validar nossa ferramenta proposta, realizamos testes utilizando o simulador S-Educ com três grupos diferentes de usuários: professores, alunos e adultos leigos. De acordo com os professores testados, o simulador S-Educ é adequado para alunos de 7 a 17 anos. Com relação à facilidade de uso da ferramenta, os professores classificaram as ações executadas durante o uso do simulador como fáceis. Os alunos também apresentaram facilidade durante a utilização do simulador, além de demonstrarem uma vontade de utilizá-lo em aulas de robótica, já que 86,49% dos alunos optaram por aulas de robótica mistas, que envolvem a utilização de simuladores e de *kits* de robótica. Por fim, o resultado obtido durante a realização de testes com os adultos leigos também foi bem sucedida, já que este grupo de usuários apresentou interesse na ferramenta, classificando a mesma como muito divertida, e apresentaram facilidade na sua utilização.

Para trabalhos futuros, pretendemos desenvolver mais algumas funcionalidades no simulador S-Educ, para implantá-lo em escolas brasileiras. Para tornar a ferramenta mais abrangente, planejamos desenvolver uma versão *online* deste simulador, permitindo que escolas do Brasil todo possam utilizar esta ferramenta em aulas de robótica educacional.

Pretendemos também modelar a velocidade de cada roda do robô separadamente, fazendo com que o movimento do robô virtual se assemelhe ainda mais ao movimento do robô real, e facilitando a realização de ações que descrevem a quantidade de rotações do motor.

Ainda, visando disseminar o seu uso e melhorar a sua interface atual, pretendemos

incluir categorias simuladas na Olimpíada Brasileira de Robótica (OBR), voltadas ao ensino fundamental e médio. Desta forma, mesmo escolas que não possuam kits de robóticas poderão participar da OBR, dando aos seus alunos a oportunidade de participar de tão importante evento. Convém ressaltar que nossas experiências com a OBR vão além, estivemos presentes em duas edições como instrutoras do curso de Duatlon, categoria voltada ao ensino médio. Esses alunos não possuem kits de robótica em suas escolas, e aprender o S-Educ e levar ele às escolas seria uma maneira de disseminar a robótica pelas escolas do Brasil afora.

Acreditamos que com o trabalho desenvolvido e ainda a desenvolver, podemos contribuir muito, atraindo mais alunos às carreiras tecnológicas, além de estarmos proporcionando melhorias no processo tradicional do ensino-aprendizagem.

Referências Bibliográficas

- AFRON (2012), ‘African robotics network’, Disponível em: <http://robotics-africa.org/>. Acesso em: 20 novembro 2012.
- Araujo, Jonathan O. (2010), ‘Desenvolvimento de jogos 3d com ferramentas gratuitas’, Trabalho de Conclusão de Curso, UNISANTOS, Santos/SP.
- Aroca, Rafael V., Renata P. Barros, Aquiles Burlamaqui & Luiz M. G. Gonçalves (2012), ‘Um robô por aluno: uma realidade possível’, *Workshop de Robótica Educacional - WRE 2012 - Fortaleza, Ceará* .
- Autodesk (2012), ‘Autodesk inventor’, Disponível em: <http://www.autodesk.com.br/>. Acesso em: 01 junho 2012.
- Barros, Renata P. (2008), ‘Roboeduc - uma ferramenta para programação de robôs lego’, Trabalho de Conclusão de Curso, UFRN, Natal/RN.
- Barros, Renata P. (2011), Evolução, avaliação e validação do software roboeduc, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, UFRN, Natal, RN.
- Becker, Guilherme L. (2010), ‘Desenvolvimento de um simulador para um veículo autônomo’, Trabalho de Conclusão de Curso, UFMG, Belo Horizonte/MG.
- Benitti, Fabiane B. V., Adilson Vahldick, Diego L. Urban, Matheus L. Kruegerand & Arvid Halma (2009), ‘Experimentação com robótica educativa no ensino médio: ambiente, atividades e resultados’, *WIE - XV Workshop sobre Informática na Escola* .
- Bisolo, Lennon R. & Carlos H. Bughi (2009), Um ambiente virtual interativo em 3d para o projeto tamar-icmbio / praia do forte - ba.
- Blender (2012), ‘Blender’, Disponível em: <http://www.blender.org/>. Acesso em: 01 junho 2012.

- Brickxcc (2012), ‘Bricx command center’, Disponível em: <http://bricxcc.sourceforge.net/>. Acesso em: 22 setembro 2012.
- Cogley, James, Paul Hempseed & Will Smith (2008), C0600 technical report - computer animation in java emergent behaviour in autonomous agents, Relatório técnico, University of Kent, Canterbury, UK.
- CyberBotics (2012), ‘Webots’, Disponível em: <http://www.cyberbotics.com>. Acesso em: 05 novembro 2012.
- DIEB (2012), ‘Dicionário interativo da educação brasileira: Educa brasil 2012’, Disponível em: <http://www.educabrasil.com.br/eb/dic/dicionario.asp>. Acesso em: 02 junho 2012.
- Fernandes, Carla C. (2010), ‘Ambiente simulado da metodologia roboeduc’, Trabalho de Conclusão de Curso, UFRN, Natal/RN.
- Fernandes, Carla, Sarah Thomaz & Luiz Marcos Gonçalves (2012a), ‘Ambiente de simulação bidimensional para robótica educacional’, *XI JORNADA PERUANA DE COMPUTACIÓN (JPC) - Puno, Peru* .
- Fernandes, Carla, Sarah Thomaz & Luiz Marcos Gonçalves (2012b), ‘S-educ: Um simulador de ambiente de robótica educacional em plataforma virtual.’, *I Workshop on MSc Dissertation and PhD Thesis in Robotics (WTDR) - Fortaleza, CE* .
- Fernandes, Carla, Sarah Thomaz & Luiz Marcos Gonçalves (2012c), ‘Uma nova abordagem em robótica educacional utilizando simuladores e kits de robótica livre’, *III Workshop de Robótica Educacional (WRE) - Fortaleza, CE* .
- Filho, Dante A. Medeiros & Paulo C. Gonçalves (2008), ‘Robótica educacional de baixo custo: Uma realidade para as escolas brasileiras’, *Anais do XXVIII Congresso da SBC - XIV Workshop de Informatica na Escola, Belém, PA, Brasil* .
- Foley, C., Van Dam, Feiner & Hughes (1996), *Computer Graphics: Principles and Practice*, Addison-Wesley.
- Fortes, Renata M. (2007), Interpretação de gráficos de velocidade em um ambiente robótico, Dissertação de mestrado, PUC, São Paulo, SP.
- Gomes, Marcelo C., Dante A. C. Barone & Ulisses Olivo (2008), ‘Kickrobot: Inclusão digital através da robótica em escolas públicas do rio grande do sul’, *Simpósio Brasileiro de Informática na Educação* .

- Gonçalves, Paulo C. (2007), Protótipo de um robô móvel de baixo custo para uso educacional, Dissertação de mestrado, Universidade Estadual de Maringá, Maringá - PR.
- Google, Armazém (2012), ‘Armazém google 3d’, Disponível em: <http://sketchup.google.com/3dwarehouse/?hl=pt-br>. Acesso em: 22 setembro 2012.
- Guzinski, Mauricio P. (2009), ‘An api architecture for representing animated conversational agents’, Trabalho de Conclusão de Curso, Unisinos, São Leopoldo/RS.
- Guzinski, Mauricio P. & Patricia A. J. Maillard (2009), Representação dos personagens de agentes conversacionais animados em 3d.
- Hoss, Alessandro, Marcelo da S. Hounsell & André B. Leal (2009), ‘Virbot4u: Um simulador de robô usando x3d’, *I Simpósio de Computação Aplicada, Passo Fundo - RS: SBC* pp. 1–15.
- JMonkeyEngine (2002), ‘Java monkey engine wiki, manual e forums’, Disponível em: <http://www.jmonkeyengine.org/>. Acesso em: 25 maio 2012.
- Jonassen, David (1996), ‘O uso de novas tecnologias na educação a distância e a aprendizagem construtivista’, *Em Aberto, Brasília, ano 16, n.70, abr/jun* .
- LDD (2012), ‘Lego digital designer’, Disponível em: <http://ldd.lego.com/>. Acesso em: 01 junho 2012.
- LEGO (2012), ‘Lego mindstorms’, Disponível em: <http://www.mindstorms.lego.com/>. Acesso em: 25 maio 2012.
- Michel, Olivier (2004), ‘Cyberbotics ltd - webotstm: Professional mobile robot simulation’, *International Journal of Advanced Robotic Systems* 1(1), 40–44.
- Miranda, Leonardo C. (2006), Robofácil: Especificação e implementação de artefatos de hardware e software de baixo custo para um kit de robótica educacional, Dissertação de mestrado, Universidade Federal do Rio de Janeiro, UFRJ, Rio de Janeiro, RJ.
- Miranda, Leonardo C., Fábio F. Sampaio & José Antonio S. Borges (2010), ‘Robofácil: Especificação e implementação de um kit de robótica para a realidade educacional brasileira’, *Revista Brasileira de Informática na Educação* 18(3).

NatalNet (2012), ‘Laboratório natalnet’, Disponível em: <http://www.natalnet.br>. Acesso em: 30 novembro 2012.

Neves-Júnior, Othon R. (2011), Desenvolvimento da fluência tecnológica em programa educacional de robótica pedagógica, Dissertação de mestrado, Universidade Federal de Santa Catarina, UFSC, Florianópolis, SC.

NXC (2012), ‘Not exactly c’, Disponível em: <http://bricxcc.sourceforge.net/nbc/nxcdoc/index.html>. Acesso em: 22 setembro 2012.

OBR (2012), ‘Olimpíada brasileira de robótica 2012’, Disponível em: <http://obr.org.br>. Acesso em: 22 setembro 2012.

Obst, Oliver & Markus Rollmann (2004), ‘Spark - a generic simulator for physical multi-agent simulations’, *Lecture Notes in Artificial Intelligence* **3187**, 243–257.

OpenGL (2012), ‘Opengl’, Disponível em: <http://www.opengl.org>. Acesso em: 22 setembro 2012.

Papert, Seymour M. (1985), *Logo: Computadores e Educação*, Editora Brasiliense, SP, Brasil.

Pedrosa, Eurico F. (2010), Simulated environment for robotic soccer agents, Dissertação de mestrado, Universidade de Aveiro, Portugal.

Pinto, Marcos (2011), AplicaÇÃo de arquitetura pedagÓtica em curso de robÓtica educacional com hardware livre, Dissertação de mestrado, Universidade Federal do Rio de Janeiro - UFRJ, Rio de Janeiro, RJ.

Prado, José P. (2008), ‘Robôs estarão disponíveis para estudantes brasileiros.’, Disponível em: <http://www.acessasp.sp.gov.br/html/modules/news/article.php?storyid=466>. Acesso em: 02 junho 2012.

RoboCup (2012), ‘Robocup 2012’, Disponível em: <http://www.robocup2012.org/>. Acesso em: 02 junho 2012.

RoboMind (2012), ‘Robomind’, Disponível em: <http://www.robomind.net/>. Acesso em: 01 junho 2012.

RVW (2012), ‘Robot virtual worlds’, Disponível em: <http://www.robotc.net/download/rvw/>. Acesso em: 01 junho 2012.

- Santos, Franklin L., Flávia M. S. Nascimento & Romildo M. S. Bezerra (2010), ‘Reduc: A robótica educacional como abordagem de baixo custo para o ensino de computação em cursos técnicos e tecnológicos’, *Anais do XVI Workshop Sobre Informática na Escola - WIE 2010, Belo Horizonte, MG, Brasil* .
- Silva, Alzira F. (2009), RoboEduc: Uma Metodologia de Aprendizado com Robótica Educacional, Tese de doutorado, UFRN, Natal, RN.
- SketchUp (2012), ‘Sketchup google’, Disponível em: <http://www.sketchup.com>. Acesso em: 22 setembro 2012.
- Steinbauer, Matthias (2007), ‘Tracking collisions in a peer-to-peer environment’, Tese de Bacharel em Ciências da Engenharia no estudo ciência da Computação, UFRN, Natal/RN.
- Sá, Sarah T. L. (2011), ‘H-educ: Um hardware de baixo custo para a robótica educacional’, Trabalho de Conclusão de Curso, UFRN, Natal/RN.
- USARSim (2012), ‘Usarsim’, Disponível em: <http://sourceforge.net/projects/usarsim/>. Acesso em: 01 junho 2012.
- VEX (2012), ‘Vex robotics design system’, Disponível em: <http://www.vexrobotics.com.br>. Acesso em: 28 setembro 2012.
- Wolf, Denis F., Eduardo V. Simões, Fernando S. Osório & Onofre T. Junior (2009), ‘Robótica móvel inteligente: Da simulação às aplicações no mundo real’, *XXIX Congresso da SBC. Jornada de Atualização em Informática* .
- Zilli, Silvana (2004), A robÓtica educacional no ensino fundamental: Perspectivas e prÁtica, Dissertação de mestrado, Universidade Federal de Santa Catarina - UFSC, Flo- rianópolis - SC.

Apêndice A

Questionário 1: Professores

1. Que disciplinas se beneficiariam na utilização do simulador S-Educ?

2. Que tipos de conteúdos você acha que consegue passar utilizando o simulador S-Educ?

3. Você acha que a utilização do simulador pode auxiliar no aprendizado em sala de aula?
() Sim () Não
4. Você achou o simulador S-Educ fácil ou complexo de utilizar?
Fácil Complexo
(1) (2) (3) (4) (5)
5. Marque a complexidade dos seguintes itens:
Classifique: Fácil Complexo
Criar atividade (1) (2) (3) (4) (5)
Montar robô (1) (2) (3) (4) (5)
Utilizar ambiente virtual (1) (2) (3) (4) (5)
6. Você achou que o menu ajuda do S-Educ pode auxiliar durante o desenvolvimento das atividades?
() Sim () Não
7. Você acha que o simulador S-Educ simula bem o movimento do robô? (apenas para professores de robótica)
Muito bem Péssimo
(1) (2) (3) (4) (5)
8. O que pode ser melhorado no simulador S-Educ?

9. Como você avalia a importância do ensino da robótica educacional no processo de ensino aprendizagem? Pode marcar mais de uma opção.

- () No interesse () No comportamento
() Na aprendizagem () Não observei
() Outras: _____

10. Para que faixa etária você avalia que o simulador S-Educ é adequado?

11. Se julgar necessário, deixe um comentário, critica ou sugestão sobre o simulador S-Educ e suas possibilidades.

Apêndice B

Questionário 2: Alunos

Para ser preenchido pelo aluno:

1. Gostou de usar o simulador de robótica?

Gostei muito Não gostei (1) (2) (3) (4) (5)

2. Considera o simulador divertido?

Muito divertido Nada divertido (1) (2) (3) (4) (5)

3. Você teve dificuldades para utilizar o simulador?

() Sim () Não

4. Marque a complexidade dos seguintes itens:

Classifique: Fácil Complexo

Montar robô (1) (2) (3) (4) (5)

Utilizar ambiente virtual (1) (2) (3) (4) (5)

5. Como é a interface do simulador?

Ótima Péssima

(1) (2) (3) (4) (5)

6. Você encontrou erros durante o uso do simulador?

() Sim () Não

7. Você acha que o simulador S-Educ simula bem o movimento de um robô? (apenas para alunos com conhecimento em robótica)

Muito bem Péssima

(1) (2) (3) (4) (5)

8. Que tipo de aula de robótica você acha mais divertida?

() Aula com simulador

() Aula com robô

() Aula com simulador e robô

9. Você teria vontade de ter mais aulas utilizando o simulador?

() Sim () Não

Para ser preenchido pelo professor:

1. Tempo para executar a atividade com o robô virtual?

2. Tempo para executar a atividade com o robô real?

Apêndice C

Questionário 3: Leigos

1. Gostou de usar o simulador de robótica?

Gostei muito Não gostei (1) (2) (3) (4) (5)

2. Considera o simulador divertido?

Muito divertido Nada divertido (1) (2) (3) (4) (5)

3. Você teve dificuldades para utilizar o simulador?

() Sim () Não

4. Marque a complexidade dos seguintes itens:

Classifique: Fácil Complexo

Montar robô (1) (2) (3) (4) (5)

Utilizar ambiente virtual (1) (2) (3) (4) (5)

5. Como é a interface do simulador?

Ótima Péssima

(1) (2) (3) (4) (5)

6. Você encontrou erros durante o uso do simulador?

() Sim () Não