



ROBÓTICA EDUCACIONAL UTILIZANDO ROBOCODE COMO FERRAMENTA DE ENSINO PARA DIMINUIR A EVASÃO DE ALUNOS NO CURSO DE ENGENHARIA ELÉTRICA.

Alanio Ferreira - allanio007@gmail.com

Universidade Federal do Ceará
Rua Estandislau Frota , S/N – Centro
CEP 62010-560 – Sobral – Ceará

Mikael do Nascimento Araújo – mikaelaraujo@gmail.com

Universidade Estadual Vale do Acaraú – UVA
Avenida da Universidade, 850 – Betânia
CEP: 62.040-370 – Sobral – Ceará

Vandilberto Pereira – vandilberto@yahoo.com.br

Universidade Federal do Ceará
Rua Estandislau Frota , S/N – Centro
CEP 62010-560 – Sobral – Ceará

Anderson Tadeu Soares de Paiva – andersonnpaiva@hotmail.com

Universidade Federal do Ceará, Departamento de Engenharia Elétrica
Bloco das Engenharias – Campus Sobral
62.010-560 – Sobral – Ceará

Resumo: *A evasão de aluno é um problema que atinge as instituições de ensino superior em geral, originando impactos de natureza social e econômica. Por esse motivo foi adotado no curso engenharia elétrica um mini curso sobre O uso da linguagem de programação java com robocode como ferramenta de ensino aprendizagem. Esta linguagem, tornou-se cada vez mais popular ao ensino de alunos nas universidades nas áreas de engenharias, pois tem mostrado eficácia no que diz respeito a incentivar ao ensino de programação com o jogo educacional de código aberto. O presente artigo apresenta um método eficiente em diminuir a grande evasão de alunos. O mini curso visa atividades relacionadas a tecnologias em diversos aspectos da área de computação: raciocínio lógico, matemático, inteligência artificial, redes neurais e programação orientada a objetos através da integração com o software Robocode. As atividades propostas foram aplicadas aos alunos do ensino superior da*



Universidade Federal do Ceará na disciplina de Introdução a Engenharia. Mostraremos também, que essas atividades tem demonstrado a eficiência da robótica educacional como ferramenta de ensino-aprendizagem em programação orientada a objetos.

Palavras-chave: *Ensino de Java, Evasão, Robocode, Programação com jogos, Robótica Educacional.*

1. INTRODUÇÃO

A técnica de ensino aplicada no Brasil no ensino superior nas Universidades é um processo de preparo de raciocínio metodológico. Atualmente existem diversas maneiras utilizadas para facilitar o ensino-aprendizagem nos ambientes educacionais. Vários trabalhos têm sido desenvolvidos, como em (Raabe, 2005; Rapkiewicz et al, 2006; Theiss, 2006; Mattos e Fuchs, 2007) objetivando da suporte ao aluno a suprir suas necessidades e obstáculos iniciais no aprendizado de programação.

As perdas de estudantes que iniciam, mas não terminam seus cursos são desperdícios sociais, acadêmicos e econômicos. No setor público, são recursos públicos investidos sem o devido retorno. No setor privado, é uma importante perda de receitas. Em ambos os casos, a evasão é uma fonte de ociosidade de professores, funcionários, equipamentos e espaço físico. (LOBO et al., 2009).

Com o objetivo de aperfeiçoar as técnicas de ensino utilizadas no Brasil evitando a evasão e, incentivar os alunos para o estudo nas áreas de engenharia, a robótica educacional aparece como uma ferramenta de ensino-aprendizagem que induz o aluno a dividir um problema, e a organização do caminho que ele deve elaborar para resolvê-lo. Dessa forma, diversos conhecimentos são difundidos nas diversas disciplinas para o estudo de engenharia, tais como: Matemática, física, inteligência artificial e programação orientada a objetos em linguagem java. Também, vale salientar que os alunos tem um pouco de contato com a língua inglesa, visto que toda a área do software incluindo menus e comandos no editor são notoriamente em inglês.

O presente artigo tem como objetivo evitar a evasão de alunos suprindo esta necessidade de permanência no ensino superior fornecendo-lhes uma integração do software *Robocode* através de atividades propostas de modo a incentivar, despertar a sua curiosidade por robótica e programação usando sua criatividade na solução de problemas. A linguagem de programação orientada a objetos através do software Robocode é bastante didática, facilitando a aprendizagem inicial do aluno. O presente artigo também apresenta algumas idéias e estratégias com resultados das atividades propostas. Será abordado como o ambiente poderia ser utilizado efetivamente como uma ferramenta de ensino, mostrando a eficiência da robótica educacional como ferramenta de ensino-aprendizagem diminuindo a evasão de alunos no curso de engenharia elétrica.

2. ROBOCODE E SUA PLATAFORMA DE ENSINO

2.1. ROBOCODE

O Robocode foi desenvolvido pela AlphaWorks, uma divisão da IBM que tem como finalidade difundir novas tecnologias de desenvolvimento. O Robocode é um projeto hospedado pela SourceForge e pode ser encontrado em <http://robocode.sourceforge.net>. A idéia básica é fornecer um ambiente de simulação de batalhas entre robôs que execute em qualquer plataforma Java. Cada tanque é controlado por um programa de computador escrito na linguagem Java, informações sobre tal linguagem podem ser obtidas em Flanagan (2000), Hagggar (2000) e Newman(1997).

Segundo Robocode (Robocode, 2008b) os robôs são programados robôs com objetivo de destruir os robôs adversários. Eles atiram, contém radares, batem uns nos outros, e movimentam-se livremente em qualquer ângulo. As batalhas de robôs acontecem no próprio Robocode em um ambiente ordenado por turnos. Dessa forma, o conceito de Robocode é fácil de entender, além de ser um jogo de programação, este é usado para aprender a programar de maneira divertida, principalmente na linguagem Java, apesar de outras linguagens como C # e Scala estão se tornando populares também. Escolas e universidades estão usando Robocode como auxiliar no ensino de programação e também para estudar a inteligência artificial (AI).

Para programar um robô, estende-se uma classe da API do Robocode. Essa API (Robocode, 2008a) é bem extensa, garantindo recursos para programar robôs guerreiros dos mais variados tipos. Existem um ambiente “Figura 2” para criar uma arena e configurar a batalha adicionando os robôs.



Figura 2 - Tela principal do ambiente virtual do Robocode

A interatividade com o usuário é programada através de eventos para tratamento do mouse e teclado. Além disso, existem eventos para avisar ao robô quando ele acerta um tiro no adversário ou é acertado, quando ele é eliminado da batalha, ou quando ela é finalizada.

O ambiente virtual apresentado na “Figura 2” contém um editor de código fonte aberto para programar os robôs. O Robocode também suporta robôs em desenvolvimento utilizando IDEs externos, como por exemplo, Eclipse, IntelliJ IDEA, NetBeans, Visual Studio, etc.

2.2 – CONHECENDO OS ROBÔS E SUA PLATAFORMA

O Robocode possui seu próprio instalador, editor interno de robô e compilador Java, o único pré-requisito é ter uma Máquina Virtual Java (JVM) instalada no sistema operacional onde ele será instalado. O fato do Robocode rodar na plataforma Java o torna possível sua execução tendo sua portabilidade em qualquer sistema operacional com Java pré-instalados, o que significa que ele será capaz de rodar em Windows, Linux, Mac OS, mas também UNIX e suas variantes. Assim, tudo que um desenvolvedor precisa para começar é fornecido com o arquivo principal Robocode distribuição (robocode-xxx-setup.jar) .

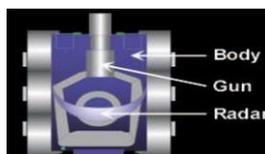


Figura 3 - Anatomia de um robô.

A anatomia do Robocode através da “Figura 3” consiste em três partes: corpo, arma, radar. Destacamos por definição as suas funcionalidades a seguir:

Corpo - Carrega a arma com o radar em cima. É utilizado para mover o robô para frente e para trás, bem como virar à esquerda ou à direita.

Arma - Montada sobre o corpo. É utilizado para disparar balas de energia. A arma pode virar à esquerda ou à direita.

Radar - Montado sobre a arma. É usado para analisar os outros robôs, pode virar à esquerda ou à direita. O radar gera eventos onScannedRobot quando os robôs são detectados.



Figura 4 - Visão do robô com radar

Existem algumas restrições em relação a energia e o calor para todos os robôs que são mencionadas a seguir:

Energia - É utilizada nas operações do robô e pode ser recuperada quando acerta-se outro robô. Durante o disparo, pode-se definir uma quantidade de energia a ser utilizada.

Calor - Um canhão só dispara quando o seu calor estiver em zero. O calor gerado é proporcional à potência do disparo.



Destacamos que as convenções de coordenadas e direção são executadas pelos seguintes ângulos de 0 à 360 graus definidos e configurados pelo programador. Na “Tabela 1” contém informações essenciais de métodos que são da prototipagem oriundas das bibliotecas do próprio editor sobre instruções para que um programador possa ter como referência sobre ações dos robôs como: comandos, ajustes de movimento e eventos.

Tabela 1: Instruções sobre métodos

COMANDOS DO ROBÔ	
<i>Método</i>	<i>Funcionalidade do método</i>
ahead (double distance)	Move o robô para frente
back (double distance)	Move o robô para trás
fire (double power)	Atira uma bala
turnLeft (double degrees)	Vira o robô para a esquerda
turnRight (double degrees)	Vira o robô para a direita
turnGunLeft (double degrees)	Vira o canhão do robô
turnGunRight (double degrees)	Vira o canhão do robô
AJUSTES DE MOVIMENTO DO ROBÔ	
<i>Método</i>	<i>Funcionalidade do método</i>
setAdjustGunForRobotTurn (boolean flag)	Se verdadeiro, vira o canhão junto com o veículo
setAdjustRadarForRobotTurn (boolean flag)	Se verdadeiro, vira o radar junto com o veículo
setAdjustRadarForGunTurn (boolean flag)	Se verdadeiro, vira o radar junto com o canhão
EVENTOS	
<i>Método</i>	<i>Funcionalidade do método</i>
onScannedRobot (ScannedRobotEvent event)	Método chamado quando o radar detecta um robô
onHitByBullet (HitByBulletEvent event)	Método chamado quando o robô for atingido por uma bala
onHitRobot (HitRobotEvent event)	Método chamado quando o robô esbarra em outro
onHitWall (HitWallEvent event)	Método chamado quando o robô atinge uma parede

A ideia principal de robôs no Robocode está concentrada num método. Cada robô é uma entidade que controla o seu próprio ciclo de vida. Ao iniciar uma batalha, um método dos robôs é invocado e disparado, e a inteligência dele é codificada nesse método. Enquanto esse método estiver sendo invocado, notoriamente o robô está andando e atirando nos seus oponentes.

No seu ambiente virtual visto na “figura 5” é bem pobre de recursos. Apesar de possuir palavras case sensitive, não contém função de auto-completar e destaque de sintaxe como é observado na NetBeans IDE. A execução do robô na plataforma virtual exige uma criação de uma arena na qual o mesmo é adicionado. A API dele é

fundamental para alunos com experiência básica em programação, pois observamos que os robôs utilizam eventos, e precisam ser passados ângulos para os movimentos.

Através da “Figura 5” temos uma versão exemplo de código do editor de uma protipagem simples. O Robocode possui dois tipos de classe que pode ser implementada: a classe simples estendida da classe Robot e a classe avançada estendida da classe AdvancedRobot. Após a instalação do software iniciamos o jogo conforme mostrado na “Figura 6” através do menu Battle, em seguida, selecionar New. Depois surgirá uma janela com informações pedindo para adicionar o nome do robô com um botão de confirmação.

```

1 package robocode_teste;
2 import robocode.*;
3 //import java.awt.Color;//API
4 //help:http://robocode.sourceforge.net/docs/robocode/robocode/Robot.html
5 public class Exemplo1 extends Robot{/* run: Exemplo1's default behavior */
6     public void run() {
7         // Initialization of the robot should be put here
8         // After trying out your robot, try uncommenting the import at the to
9         while(true) {
10            ahead(100); turnGunRight(360);
11            back(100); turnGunRight(360);
12        }
13    }
14    /*onScannedRobot: What to do when you see another robot */
15    // Replace the next line with any behavior you would like
16    public void onScannedRobot(ScannedRobotEvent e) {
17        fire(1);
18    }
19    /* onHitByBullet: What to do when you're hit by a bullet */
20    // Replace the next line with any behavior you would like
21    public void onHitByBullet(HitByBulletEvent e) {
22        back(10);
23    }
24    /* onHitWall: What to do when you hit a wall */
25    // Replace the next line with any behavior you would like
26    public void onHitWall(HitWallEvent e) {
27        back(20);
28    }
29 }
30
    
```

Figura 5 - Editor de código do Software Robocode

Logo em seguida após essa confirmação teremos uma nova janela pedindo o nome do pacote com a confirmação para mostrar a janela de editoração. O próximo passo é desenvolver a implementação do código no editor visto na “Figura 5”, ou desenvolver um novo robô com outras aplicações e derivações de classes e métodos dando assim uma melhor robustez. Após o código do robô já implementado precisamos compilar através do menu compiler, em seguida para salvar usa-se como referência a “Figura 6” clicando no menu Battle, logo depois usamos o botão salvar e escolhendo na outra janela um nome para o arquivo.

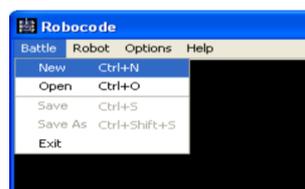


Figura 6 - Começando um jogo

O Robocode possui uma grande quantidade de robôs demo em que o programador pode usar sua estrutura de código como referência de estudo para utilizar a



tecnologia de código já programada da própria prototipagem do software para implementar seu próprio robô.

A batalha se inicia após adicionarmos os robôs na tela para adicionar os robôs a competição usamos o menu Battle, em seguida New, logo depois irá aparecer uma nova janela contendo os robôs que queremos adicionar a uma batalha escolhendo o tipo de robô e clicando no botão Add. Também é possível configurar nessa mesma janela através do menu Rules o tempo e número de Rounds. Após o término da batalha é apresentado uma tela com pontuação descrevendo detalhes do vencedor sobre: total de score, rank e etc. Também contém uma tela para adicionar os robôs a competição com três tipos de pacotes em sua representação, caso o aluno ou o programador venha a criar futuramente um robô, este por sua vez, será apresentado nessa mesma janela. Os pacotes são representados para identificar o tipo da classe pertencente aos seus robôs que são representados por: 16 robôs do pacote da classe sample, 5 robôs do pacote da classe simplex, 3 robôs do pacote sampleteam. Demais pacotes que possam surgir são criados pelo próprio programador.

3. O MINI CURSO

O mini-curso foi lecionado para 50 alunos matriculados na disciplina de Introdução a Engenharia do primeiro semestre do curso de Engenharia Elétrica da Universidade Federal do Ceará, *Campus Sobral*. O mini-curso teve duração de dois dias, cada dia com 2 horas de aula. O cronograma do mini curso é mostrado na tabela abaixo:

Tabela 2: Cronograma do mini curso.

PRIMEIRO DIA	CONTEÚDO	SEGUNDO DIA	EXERCÍCIOS
Teoria	Sobre o Mini Curso	Prática	Avançar para frente
	O que é o Robocode?		Virar o canhão com angulação de até 360°
	Conceitos Básicos		Andar para traz
	Anatomia do Robô		Estratégia para fugir dos tiros
	A Batalha		Estratégia após ser atingido por disparos
	A Competição		Estratégia após ser detectado por radares
	Como Instalar		Estratégia após detectar as paredes
	Métodos Básicos		
	Eventos		
	Criando um Robô		
	Exemplo de Código		
	Iniciando uma Batalha		

O mini curso foi realizado no laboratório de informática do *Campus*, pelo bolsista e seu orientador com a participação do professor que lecionava a disciplina de Introdução a Engenharia. O robô construído para o desenvolvimento de atividades em



sala de aula é mostrado seu código na “Figura 7” possuindo problemas que induzam à procura pelos recursos necessários para resolvê-los, através de uma estratégia conhecida como PBL (Problem Based Learning ou Aprendizagem Baseada em Problemas).

Vale salientar que as soluções apresentadas pelo bolsista, inicialmente, teve como objetivo expor os conhecimentos básicos de comandos. Após ser apresentados os comandos básicos os alunos ficam preparados para conhecer novos desafios e solucionar.

Neste contexto, os alunos estarão aptos para criar robôs e personalizar, também pesquisar comandos em fontes na internet e apresentar soluções diferenciadas de códigos, mas com o mesmo lema: construa o melhor e destrua o resto.

```

1 package meuPacote;
2 import robocode.*;
3 import java.awt.Color;
4 public class karoto extends Robot{
5     public void run() {
6         setColors(Color.orange, Color.blue, Color.yellow);
7         while(true) {
8             double i = (double) (40+Math.random()*400);
9             double j = (double) (40+Math.random()*400);
10            double k = (double) (1+Math.random()*180);
11            ahead(i); back(j); turnRight(k);
12            turnGunRight(360); setAdjustGunForRobotTurn(true);
13        }
14    }
15    public void onScannedRobot(ScannedRobotEvent e) {
16        fire(5);
17    }
18    public void onHitByBullet(HitByBulletEvent e) {
19        ahead(50); back(80);
20    }
21    public void onHitRobot(HitRobotEvent e) { back(30); }
22    public void onHitWall(HitWallEvent e) {
23        double x = getX(); double y = getY();
24        double m = getBattleFieldWidth();
25        double n = getBattleFieldHeight();
26        if((x==m) || (x==0))
27        { ahead(100); }
28        if((y==n) || (y==0))
29        { back(40); }
30    }
31 }

```

Figura 7 – Protótipo criado em sala de aula.

Mas quando os estudantes não foram capazes de resolver uma atividade de pequena ou grande complexidade, o bolsista mostrou a solução e discutiu com eles a fim de difundir conhecimento, aprendizagem com as novas soluções apartir das dificuldades apresentadas no exercício.

4 – AVALIAÇÕES E RESULTADOS

Ao final do último dia do mini-curso, os alunos responderam a um formulário de perguntas que avaliaram a eficiência deste mini-curso em relação ao despertar ou reforçar a participação dos alunos no curso de Engenharia Elétrica, levando-os a permanecer no curso. Para confirmar ainda mais a eficiência do projeto, o número de reprovados foram listados em anos anteriores e foram comparados estes números com o número de reprovados no ano avaliado (2013). A “Tabela 3” mostra as perguntas do formulário, junto com as respostas em percentuais dos estudantes:

Tabela 3: Questionário com as respostas dos alunos.

QUESTÕES	SIM	NÃO
Você ficou interessado em robótica através deste mini-curso?	67,00%	33,00%
As palestras são fácil entendimento?	80,00%	20,00%
Você recomenda esse minicurso para outros alunos?	80,00%	10,00%
O mini-curso lhe motivou em aprender mais sobre robótica?	69,00%	31,00%
Você já ouviu falar das Olimpíadas de Robótica?	54,00%	46,00%
Você já participou de alguma olimpíada de robótica?	9,00%	91,00%
Você tem interesse em um treinamento para participar de competições de robótica?	64,00%	36,00%
Você já teve algum contato com a Robótica antes do mini-curso?	11,00%	89,00%
Você já conhecia o software Robocode antes do mini-curso?	7,00%	93,00%

Percebemos que baseado na “Tabela 3”, que a maioria dos estudantes recomendariam este mini-curso para outros alunos. Em percentual maior por parte da classe, essa foi capaz de resolver e conhecer as funcionalidades somente nas primeiras cinco atividades sem problemas, no entanto, apenas um pequeno grupo de alunos foi capaz de pesquisar e resolver duas últimas atividades: estratégia após ser detectado por radares, estratégia após detectar as paredes do ambiente.

Isso mostra que o mini-curso estava em um nível intermediário para o primeiro semestre de um curso de Engenharia Elétrica. Foi realizado um levantamento das disciplinas básicas de engenharia (Cálculo Diferencial e Integral I, Física Geral I e Programação Computacional) oferecidos no período de 2006 à 2013. Ao fazermos uma análise de acordo com a “Tabela 4”, observa-se que o número de matriculados em 2013 é muito maior do que o número de matriculados nos anos anteriores e, que a percentagem de alunos que foram reprovados nessas disciplinas que indiretamente envolvem a robótica é relativamente menor em 2013, do que a percentagem nos anos anteriores (2006 à 2012).

De forma geral, o curso de Robocode foi aplicado pela primeira vez em 2013, em que a quantidade de inscritos foi compatível em maior número em relação a outros mini cursos ministrados em anos anteriores, podemos visualizar que o mini-curso realmente fez os alunos possuírem motivação para optarem em seguir o curso de Engenharia Elétrica. A eficiência deste mini curso é mostrada tanto na “Tabela 3”, com o formulário que foi aplicado imediatamente após o mini-curso, como também na “Tabela 4”, na qual se compararmos com os anos anteriores, observamos que o número de alunos que reprovaram em 2013 é um dos menores índices.

Tabela 4: Percentagem de alunos reprovados.

Ano	Cálculo Diferencial e Integral I			Física Geral I			Programação Computacional		
	Mat*	Rep*	%	Mat*	Rep*	%	Mat*	Rep*	%

2006	34	14	41,18%	34	14	41,18%	34	24	70,59%
2007	42	30	71,43%	34	22	64,71%	53	45	84,91%
2008	50	33	66,00%	63	33	52,38%	70	43	61,43%
2009	53	39	73,58%	57	34	59,65%	47	27	57,45%
2010	70	41	58,57%	66	52	78,79%	51	33	64,71%
2011	83	29	34,94%	102	43	42,16%	101	42	41,58%
2012	56	44	78,57%	52	28	53,85%	50	27	54,00%
2013	108	48	44,44%	111	25	22,52%	112	52	46,43%

Mat*=matriculados; Rep*=reprovados;

O baixo rendimento acadêmico nestas disciplinas introdutórias dos cursos de engenharia pode ser uma das principais causas da evasão no curso de Engenharia Elétrica e retenção de alunos nos semestres iniciais. Esta falha pode ser atribuída à má formação do aluno durante o ensino nas escolas de ensino médio (SILVA *et al*, 2012). Podemos entender melhor com os gráficos: “Figura 8”, “Figura 9”, “Figura 10” abaixo. Esta por sua vez, expõe o baixo percentual de reprovados nas principais disciplinas básicas de engenharia em 2013, em relação aos anos anteriores.

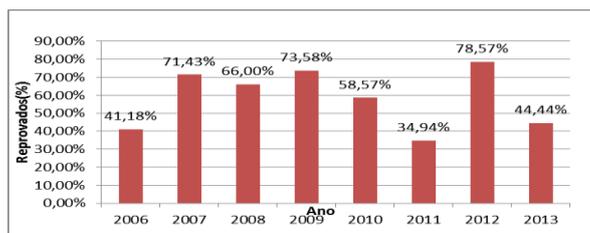


Figura 8 - Reprovados em Cálculo Diferencial e Integral I.

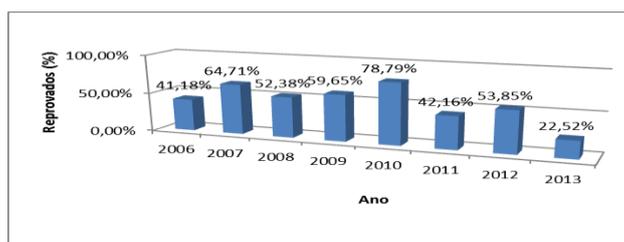


Figura 9 - Reprovados em Física Geral I.

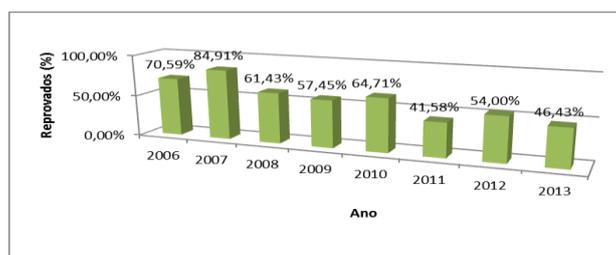


Figura 10 - Reprovados em Programação Computacional.

5. CONCLUSÕES E TRABALHO FUTURO

As atividades aplicadas e desenvolvidas no presente artigo facilitam o ensino e a aprendizagem dos alunos no primeiro semestre e os incentivam sua permanência no curso. Os alunos estavam muito curiosos durante e após a realização das atividades, tal fato é evidenciado na Tabelas 3 e Tabelas 4, que nos mostram o caráter motivacional da robótica como ferramenta de ensino.

O Bolsista facilitador observou que as atividades envolvendo a programação do software Robocode é válida para aqueles que desejam aprender ou aprimorar seus conhecimentos em linguagem de programação orientados a objetos. Percebemos que por meios de dados estatísticos expostos nas tabelas e gráficos mostrados neste artigo que o mini-curso deixou os alunos motivados a continuarem no curso e, reduziu o número de reprovações em disciplinas introdutórias essenciais para a engenharia. Também que visa reduzir a evasão de alunos em cursos de Engenharia Elétrica.

Por fim o que foi abordado neste trabalho pode auxiliar estudantes e professores na área de programação e robótica, pois possui a implementação de vários conceitos importantes para estas áreas e pode servir como base para o desenvolvimento de trabalhos futuros.

REFERÊNCIAS BIBLIOGRÁFICAS:

Raabe, A. L. A. (2005) "**Uma proposta de arquitetura de Sistema Tutor Inteligente baseada na teoria das experiências de aprendizagem mediadas**". Tese (Doutorado) – Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre.

Theiss, F. (2006) "**Linguagem visual orientada por figuras geométricas voltada para o ensino de programação: versão 2.0**". Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

Rapkiewicz, C. E. et al. (2006) "**Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais**". Disponível em: <<http://www.cinted.ufrgs.br/renote/dez2006/artigosrenote/25157.pdf>>. Acessado em: 18 mar. 2014.

Mattos, M. M.; Fuchs, J. F. (2007) "**Qualifica: Uma ferramenta para apoio a construção de algoritmos estruturados**". In XVI Seminário de Computação - SEMINCO, Blumenau, 2007. p. 75-87.

BENITTI, F.B. VAVASSORI; VAHLDICK, A; URBAN, D.L; KRUEGER, M.L; HALMA, A: **Experimentação com Robótica Educativa no Ensino Médio: CSBC** - XXIX Congresso da Sociedade Brasileira de Computação, UFRGS, 2009.

SILVA, J.D.A.; TEIXEIRA, V.S.C.; MOREIRA, A.B.; TEIXEIRA, I.S.C.



Análise do Curso de Engenharia Elétrica do Campus de Sobral: Promover o Interesse de Estudantes do Ensino Médio e o Controle da Evasão. XL – Congresso Brasileiro de Educação em Engenharia. Belém: UFPA, 2012.

FLANAGAN, David. Java: **o guia essencial**. Rio de Janeiro: Campus, 2000.

HAGGAR, P. Java - **Guia prático de programação**. Rio de Janeiro: Campus, 2000.

NEWMAN, A. **Usando Java**. Rio de Janeiro: Campus, 1997.

Robocode (2008a) "**Robocode API**". Disponível em: <http://robocode.sourceforge.net/docs/robocode> - Acessado em: 18 mar. 2014.

Robocode (2008b) "**Robocode Home**". Disponível em: <http://robocode.sourceforge.net/> Acessado em: 18 mar. 2014.

LOBO, Roberto Leal; FILHO, Silva, LOBO, Maria Beatriz, HIPÓLITO, Oscar – **Evasão no Ensino Superior: Causas e Remédios** – Junho, 2009. Disponível em: <http://robertolobo.com.br/index.php/2009/06/evasao-no-ensino-superior-causas-e-remedios/> - Acessado em: 18 mar. 2014.

EDUCATIONAL ROBOTICS USING ROBOCODE AS A TEACHING TOOL TO DECREASE THE EVASION OF STUDENTS IN ELECTRICAL ENGINEERING COURSES

Abstract:

The student evasion is a problem that affects institutions of higher education in general, causing impacts of social and economic nature. For this reason it was adopted in electrical engineering course a mini course on using the java programming language with robocode as a tool for learning education. This language, has become increasingly popular to teaching students in universities in the areas of engineering, because it has shown effectiveness with regard to encourage the teaching of programming with the open source educational game. This paper presents an efficient method to reduce the great escape of students. The mini course aims activities related to technologies in various aspects of computing: logical reasoning, mathematics, artificial intelligence, neural networks, and object-oriented programming through the integration with the Robocode software. The proposed activities were applied to students of higher education of the Universidade Federal do Ceará in discipline of introduction to Engineering. We also believe that these activities have shown the efficiency of educational Robotics as a tool for teaching and learning in object-oriented programming.

Keywords: Teaching Java, Evasion, Robocode, Programming with games, Educational Robotics.