

Torneios Baseados em Robocode para Incentivar Jovens a Aprender Programação

Matheus C. Meira¹, Marri S. S. Lima¹, Marcos A. F. Borges¹

¹Faculdade de Tecnologia – Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 456 – 13.484-332 – Limeira – SP – Brasil

m108479@dac.unicamp.br, {m118075, marcosborges}@ft.unicamp.br

Abstract. *Students from the current generation, inserted in a context of connectivity in the digital world may consider demotivating the traditional learning environment. This article points a new approach to teaching programming through the development of scientific competitions based on the digital educational game Robocode. The study presents the experience of organizing scientific competitions involving contests and leagues based on the educational game Robocode over five years and it discuss how these can be used as promising tools to diversify playful and collaborative learning methods.*

Resumo. *Alunos das gerações atuais, inseridos em um contexto de conectividade no universo digital, podem considerar ambientes de ensino tradicionais desmotivadores. Este artigo indica uma nova abordagem para ensino de programação, a partir do desenvolvimento de competições científicas baseadas no jogo digital educacional Robocode. O artigo apresenta a experiência de organização de competições científicas que envolvem torneios e ligas baseadas no jogo educacional Robocode ao longo de cinco anos e discute como eles podem ser usados como promissoras ferramentas de diversificação de métodos lúdicos e colaborativos de aprendizagem.*

1. Introdução

Atualmente, instituições de ensino desenvolvem ações com jogos educacionais digitais nos processos de ensino-aprendizagem para incentivo ao aprendizado de programação (Barnes, Richter, Powell, Chaffin, e Godwin, 2007; Chaffin, Doran, Hicks e Barnes, 2009). Conteúdos de lógica de programação ou programação em si não caracterizam apenas assuntos exclusivos aos cursos específicos das áreas relacionadas a informática ou engenharias. Estes assuntos tornaram-se tópicos essenciais para melhor entendimento de aplicativos de uso geral, ou mesmo como ferramentas de auxílio no ensino de variadas abordagens curriculares. A programação ganha o interesse dos jovens, envoltos em tecnologias cada vez mais presentes em suas vidas. A programação consegue despertar a criatividade das pessoas que a utilizam (Valente, 1999; Garcia, Correia e Shimabukuro, 2008). Alunos participantes de disciplinas de programação ou jovens admiradores de tecnologias, muitas vezes têm o anseio de entender a lógica aplicada nos aplicativos atuais.

A organização sem fins lucrativos Code.org¹ é dedicada a expandir o ensino de ciência da computação, incentivando o acesso ao conhecimento de programação entre os jovens. Conhecido no Brasil como “hora do código”, Code.org se enquadra nas ações que têm por objetivo permitir e incentivar a compreensão de conceitos de programação de modo lúdico. O presente artigo aborda ações para incentivar jovens programadores com a diversificação de métodos de aprendizagem com auxílio de torneios científicos.

Enquanto o “método tradicional de ensino é centrado no professor, que possui o papel de sujeito ativo, sendo o aluno passivo no processo de aprendizagem” (MEZZARI, 2011, p. 115), algumas iniciativas procuram disseminar a programação com propostas de ensino que diferem aos meios tradicionais.

Estudos empíricos apontados por Johnson (2010) e Adams (2012) demonstram que os alunos podem alcançar ganhos significativos de aprendizagem ao interagir com jogos educacionais em ambientes controlados. “Jogos educacionais digitais vão muito além do entretenimento e funcionam como eficientes ferramentas de aprendizagem” (GALLANT e MAHMOUD, 2008, p. 119). “Os jogos educacionais têm sido frequentemente sugeridos como promissoras ferramentas pedagógicas” (PRENSKY, 2012, p. 22).

Este artigo busca discutir abordagens interessantes para incentivar a aprendizagem de programação entre jovens. Na tentativa de responder a esse questionamento são levantadas hipóteses no sentido de adoção de jogos educacionais digitais que possam despertar o interesse em um ambiente que proporcione competições científicas. A proposta apresentada neste trabalho explora a adoção do jogo educacional digital denominado Robocode (2013) associado ao Aprendizado Baseado em Problemas (*Problem Based Learning* - PBL) no desenvolvimento de competições científicas. PBL é uma forma de aprender com a utilização de cenários que envolvem problemas da vida real. “É um método que desafia os alunos a aprender e trabalhar em grupos na busca de soluções para os problemas reais” (HOU, 2014).

O jogo Robocode permite o ensino de linguagem de programação Java ou C# (plataforma .NET). A partir do jogo Robocode é possível organizar competições científicas que possam favorecer acesso ao estudo de disciplinas relacionadas à lógica de programação e programação de computadores de modo lúdico. O objetivo da presente abordagem está centrado na apresentação do jogo educacional Robocode em competições para estímulo de aprendizagem de linguagem de programação. As seções seguintes apresentam o Robocode e estratégias para explorar esta plataforma em competições planejadas para um ambiente de ensino-aprendizagem de programação.

2. Framework de Simulação “Robocode”

O projeto Robocode é um software *open source*, com fontes abertas e sob os termos da *Eclipse Public License*² (EPL), que possibilita simulações de batalhas com robôs virtuais. Foi projetado para simular uma arena de batalha com robôs, cada qual

¹ Code.org. (2016) *Non-profit Dedicated to Expanding Access to Computer Science*. Disponível em: <<https://code.org/>>.

² EPL. (2016) *Eclipse Public License version 1.0, definitions homepage*. Disponível em: <<http://www.eclipse.org/legal/epl-v10.html>>.

com seus próprios algoritmos de ataque e defesa. O software pode ser executado em plataformas com suporte às versões recentes do Java Runtime (versão 2 ou superior) (Robocode, 2013).

Robocode foi desenvolvido por uma comunidade da IBM® denominada *AlphaWorks*, ativa desde 1996, com propostas de desenvolvimento de tecnologias emergentes. Robocode ganhou notoriedade nos laboratórios de pesquisa da IBM®, porém com incertezas de sua continuidade, o projeto foi levado em 2005 para o *SourceForge* (consiste em um repositório de código fonte que atua como centro para desenvolvedores gerenciarem projetos livres). Atualmente, o projeto Robocode é mantido por iniciativas de software livre em torno da comunidade Robocode.

Robocode (2013) é um jogo de programação com objetivo de codificar batalhas para competir em uma arena com outros robôs. O jogador é o programador do robô virtual, cuja função é desenvolver a lógica com informações de comportamento e reação a eventos que ocorrem na arena. O software é composto por um ambiente completo de desenvolvimento: instalador próprio, editor de robôs e compilador Java/C#.

O Robocode possui uma IDE (*Integrated Development Environment*) própria e uma interface formada por componentes GUI (*Graphical User Interface*), também conhecido por *widgets* (controles para simplificar o acesso a um outro programa ou sistema). A Figura 1 indica os componentes do sistema Robocode, onde é possível visualizar as janelas de campo de disputa (*Battlefield*) e editor do robô (*Robot Editor*) que representam a forma do IDE (Li, 2002).

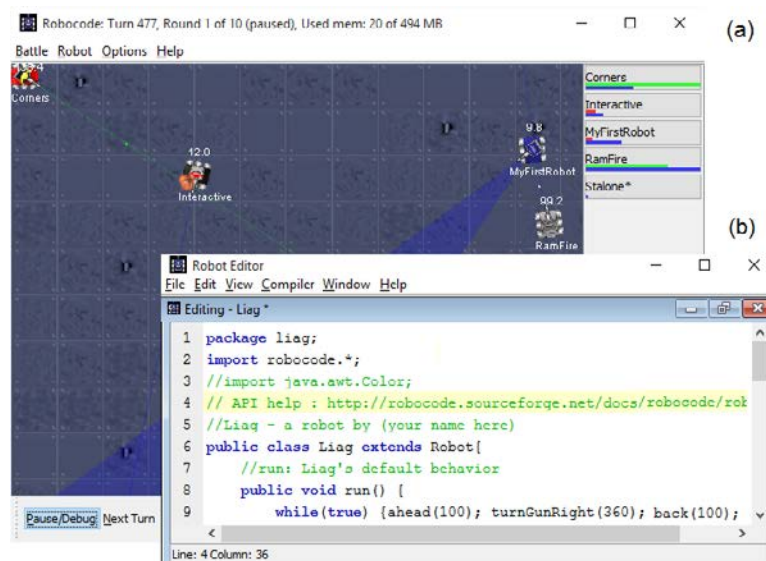


Figura 1 - Robocode (a) Campo de Batalha (b) Editor de Robô (Robocode, 2015).

Pode-se classificar Robocode como um jogo sério (*Serious Games* - SGs), por ser um jogo que proporciona um ambiente em que o agente de aprendizagem pode interagir e aprender. De acordo com Nidorf e Barone (2010), para este cenário, o agente tem a possibilidade de escolher uma variedade de comportamentos e medir a eficácia de cada um ao longo do desafio.

Com base em Recchia et al. (2014, p. 90) pode-se classificar o Robocode como um *framework* de simulação. A classificação do Robocode está centrada em um simulador de batalhas com base em Inteligência Artificial (IA). No simulador de

batalhas, é possível observar o comportamento dos robôs. Pelo fato das batalhas ocorrerem de modo *on-line*, na plataforma, é possível comparar a programação de robôs efetuada por outros desenvolvedores. No simulador, os robôs são entidades autônomas e os programadores não têm controle das ações, o que contribui para programação de agentes inteligentes. Um robô não possui total conhecimento do meio ambiente de batalha e suas informações estão restritas aos sensores do qual ele pode ler (ex. posição do robô, ângulo do canhão e/ou radar etc.) (Hong e Cho, 2004; Fayek e Farag, 2014).

3. Ambiente de Competição com Robocode

Um gerenciador da liga mundial Robocode denominado *RobotLeague* foi criado por Christian Schnell, em 2002. O gerenciador tinha funções de garantir o agrupamento dos participantes; conduzir partidas; gerenciar os resultados e produzir relatórios de status em páginas compartilhadas na internet. *RobotLeague* foi descontinuada um ano após sua criação, porém ganhou notoriedade por representar a primeira iniciativa de liga mundial do Robocode (Li, 2002). Atualmente, destaca-se a competição mundial chamada *Hat League*, mantida pela comunidade Robowiki³ associada a iniciativa colaborativa denominada *RoboRumble* para exibir as partidas e construir rankings.

Existem vários torneios Robocode no mundo, em países como: Tailândia, Bélgica, Austrália e Nova Zelândia. Dentre disputas ao redor do mundo, é possível citar como destaque o torneio da Irlanda, *GamesFleadh*⁴. O torneio da Irlanda foi iniciado pela ICS (*Irish Computer Society*), em 2006, sendo denominado “*Robocode Challenge Trophy*” e contando com a participação de estudantes de todo país. Cursos de informática (em nível médio e superior) e afins criam grupos de pesquisa e treinam suas equipes formadas por alunos dos primeiros anos das disciplinas de programação. O torneio da Irlanda proporciona uma emocionante competição com envolvimento das instituições acadêmicas do país. O torneio conta com importantes incentivadores dos eventos ao longo dos anos: Microsoft (atual); EA–*Electronic Arts* (atual); Dell; Pearson *Education*; Lenovo, etc. Originalmente, essa competição era chamada de “*Robocode Challenge Trophy*”, mas atualmente é denominado “*GamesFleadh*” (festival dos jogos na tradução da língua Irlandesa), organização mantenedora e coordenadora das disputas.

No Brasil, instituições como a ETEC (Escola Técnica Estadual - Centro Paula Souza) possuem torneios descentralizados e estudos para organização de liga em suas unidades espalhadas no estado de São Paulo. Institutos Federais possuem históricos de realizações de disputas de programação com Robocode. Várias universidades (públicas e privadas) espalhadas no Brasil já tiveram iniciativas com o torneio dentro das disciplinas de programação ou em semanas de ciência e tecnologia. A partir de pesquisas de informações registradas em domínios oficiais institucionais, a Tabela 1 exibe algumas instituições brasileiras que realizaram torneios Robocode ao longo dos últimos anos. São apresentadas apenas entidades com mais de um ano de incidência; aproximadamente 40 outras foram identificadas com apenas um ano de torneio. A classificação segue em ordem decrescente de incidências. A pesquisa foi realizada no

³ Robowiki. (2016) *Collecting Robocode Knowledge Since 2003*. Disponível em: <<http://robowiki.net>>. Código fonte aberto, com a promoção de competições ao redor do mundo.

⁴ GamesFleadh. (2016) *Digital Games Programming Festival - Robocode*. Disponível em: <<http://gamesfleadh.ie/robocode>>.

final do ano de 2015 no site de buscas do Google (2015) em domínios brasileiros com as seguintes expressões: “Torneio Robocode”; “Liga Robocode”; “Campeonato Robocode” e “Competição Robocode”.

Números apurados no presente artigo, indicam que competições disputadas entre algumas das primeiras instituições listadas na Tabela 1, Colégio Técnico de Limeira da Universidade Estadual de Campinas (COTIL-UNICAMP), Faculdade de Tecnologia da Universidade Estadual de Campinas (FT-UNICAMP) e Instituto Federal de Educação Ciência e Tecnologia de São Paulo (IFSP), caracterizam a maior competição Robocode oficial já realizada no Brasil. Esta competição ultrapassou a marca do Instituto Federal de Alagoas (IFAL) em número de participantes (60) equipes (16) e instituições envolvidas (1). Estas instituições deram origem a competição anual brasileira do Robocode organizada pelo laboratório de Informática Aprendizagem e Gestão (LIAG) da UNICAMP.

Tabela 1 - Ocorrências Torneio Robocode Brasil.

Instituição	Cidade	Ano(s) do Torneio(s)
COTIL-UNICAMP - Colégio Técnico de Limeira	Limeira-SP	2015, 2014, 2013, 2012, 2011, 2010
FT-UNICAMP - Faculdade de Tecnologia	Limeira-SP	2015, 2014, 2013, 2012, 2011, 2010
ETEC Centro Paula Souza - João Belarmino	Amparo-SP	2014, 2013, 2012, 2011
IFSP - Instituto Federal São Paulo	Capivari-SP	2015, 2014, 2011
IFAL - Instituto Federal de Alagoas	Arapiraca-AL	2015, 2014
IF Farroupilha	São Borja-RS	2015, 2014
UFU - Universidade Federal de Uberlândia	Patos de Minas-MG	2013, 2012
UFMA - Universidade Federal do Maranhão	São Luís-MA	2014, 2009
UFC - Universidade Federal do Ceará	Quixadá-CE	2011, 2007

3.1. Competições Robocode na UNICAMP e IFSP

O torneio Robocode do laboratório LIAG da UNICAMP ocorre desde de 2010. Os torneios Robocode LIAG são regidos por regras de modo estimular a competitividade, administrar as rodadas e garantir a organização. Algumas das principais regras são: (1) Equipes de no máximo quatro participantes; (2) Compostos por uma fase classificatória de pontos corridos e outra com enfrentamento direto em quadrangulares (quartas de final, semifinal e final); (3) Ao final de cada rodada, os códigos fontes dos robôs são disponibilizados e divulgados pela organização do evento; (4) Mudanças nos códigos devem ser explicadas para atender boas práticas de programação; (5) Grupos com códigos copiados são banidos da competição.

A partir de 2012, o LIAG passou a organizar Ligas. O conceito da liga é promover torneios descentralizados seguindo o mesmo padrão de regras e organizar uma final da liga entre os melhores colocados dos torneios.

No início do segundo semestre de 2015, o LIAG promoveu uma liga entre três instituições (FT-UNICAMP, COTIL-UNICAMP e IFSP) regidos sob as regras da liga. O LIAG promoveu os torneios na FT-UNICAMP e COTIL-UNICAMP com a condução e supervisão de integrantes e colaboradores do LIAG, sendo o primeiro aberto, contando com a participação de equipes da FT-UNICAMP e instituições externas e o segundo fechado apenas para alunos do COTIL-UNICAMP. Professores e alunos bolsistas de projetos relacionados ao Robocode do IFSP administraram um torneio próprio. Todas

informações das disputas (notícias dos torneios e liga, instituições participantes, resultados, ranking, disputas, premiações, cursos, fotos, vídeos, etc.) são centralizados e disponibilizados na internet em um canal oficial do evento, RobocodeBrasil⁵. Foram criados calendários distintos para cada um dos torneios locais onde cada torneio intermediário registra um campeão com direito a disputar a fase final da liga.

No final do segundo semestre de 2015, o LIAG recebeu as equipes campeãs dos torneios para organização e realização da liga entre os finalistas Robocode. Contou com a participação de equipes vencedoras do torneio do Instituto Federal Farroupilha (IF Farroupilha) do campus de São Borja-RS, disputado junto ao torneio da FT-UNICAMP. No ano de 2015, os eventos contabilizaram a participação de aproximadamente 30 equipes e mais de 100 competidores, em três torneios intermediários concomitantes e uma liga interinstitucional.

A liga dos campeões é composta por desafios diretos em nível de semifinal e final com os campeões de cada torneio intermediário. Os segundos melhores colocados de cada instituição tiveram a oportunidade de participar de uma “repescagem” e entrar no quadrangular final em busca do título de campeão.

Os torneios e a liga podem ser patrocinados e oferecer incentivo em forma de premiações. A liga LIAG 2015 contou com incentivos: (1) Destinados aos campeões dos torneios intermediários; (2) Para os três primeiros colocados da liga e; (3) Melhor código dentre todas equipes participantes. O LIAG promoveu a liga e incentivou as equipes intermediárias oferecendo cursos especializados para as mais bem colocadas.

4. Discussão

Foram respondidos 22 questionários individualmente por alunos participantes dos torneios e ligas Robocode nos anos de 2014 e 2015. O grupo foi composto por estudantes de cursos técnicos e graduação com idades que variaram de dezesseis a vinte e cinco anos e participação aproximada de dois terços do gênero masculino. A Figura 2 (a) exibe a porcentagem dos alunos regularmente matriculados que responderam questões referente as competições científicas nas instituições: FT-UNICAMP; COTIL-UNICAMP e IFSP, dentre os cursos: Técnico em Informática (médio); Técnico Integrado em Informática (médio integrado); Análise e Desenvolvimento de Sistemas (superior); Construção de Edifícios (superior) e Engenharia de Telecomunicações (superior). Algumas informações foram solicitadas aos alunos acerca do conhecimento ou entendimento da linguagem de programação Java, observado na Figura 2 (b). Os resultados apurados na Figura 2 puderam apresentar um público de alunos predominantemente da área de exatas e em sua maioria com conhecimento superficial ou apenas noção da existência da linguagem de programação Java. Alunos que indicaram o conhecimento prévio foram, em maioria, aqueles concluintes de disciplinas relacionadas a linguagem de programação.

As questões puderam envolver os alunos com objetivo de apresentar características das quais torneios científicos podem auxiliar na aprendizagem da linguagem de programação (Figura 3). Como resultado foi possível observar que cerca

⁵ RobocodeBrasil (2016) canal oficial competição. Disponível em <<http://www.robocodebrasil.com.br>>. Página acadêmica Robocode do LIAG, disponível em <<http://www.ft.unicamp.br/liag/robocode>>.

de metade das respostas focam: aprendizagem de maneira lúdica; torneio Robocode desperta a motivação e desafia os alunos a apresentar propostas para desenvolvimento dos robôs; a formação equipes colaborativas no desenvolvimento dos códigos e estratégias.

A pergunta final do questionário indica quais melhorias e iniciativas o evento Robocode deve oferecer para conquistar participação de novos alunos. Com esse foco em conquistar novos adeptos à competição Robocode, as principais sugestões foram apresentadas na Figura 3. Conforme respostas dos questionamentos, um fator considerado para o aumento do interesse por esse tipo de torneio é a mobilização de professores e alunos para disponibilizar minicursos ou tutorias para auxílio no desenvolvimento dos códigos entre as equipes. Outro fator está relacionado ao investimento e divulgação das instituições na realização das competições que envolvem jogos digitais e diversificação de métodos de ensino.

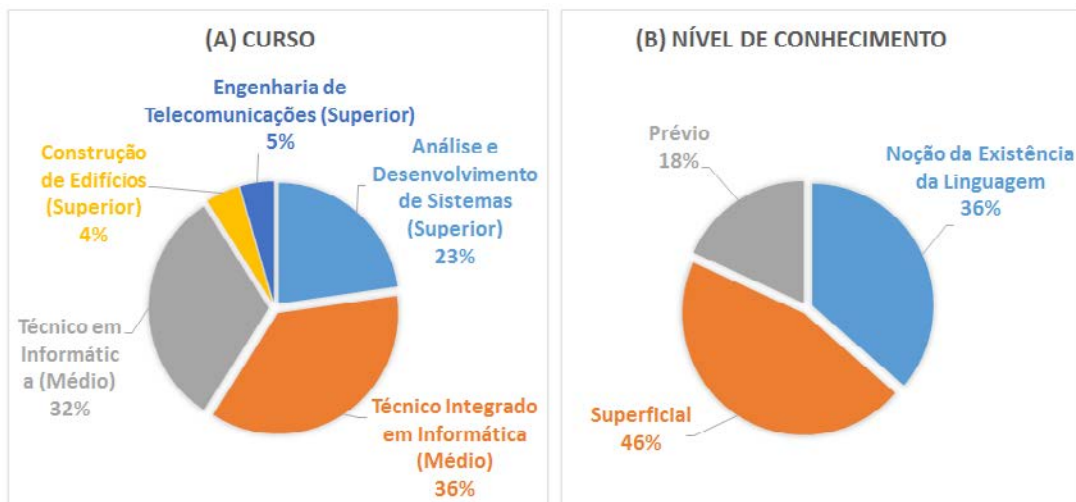


Figura 2 - (a) Participantes de Alunos por Curso (b) Nível de Conhecimento.

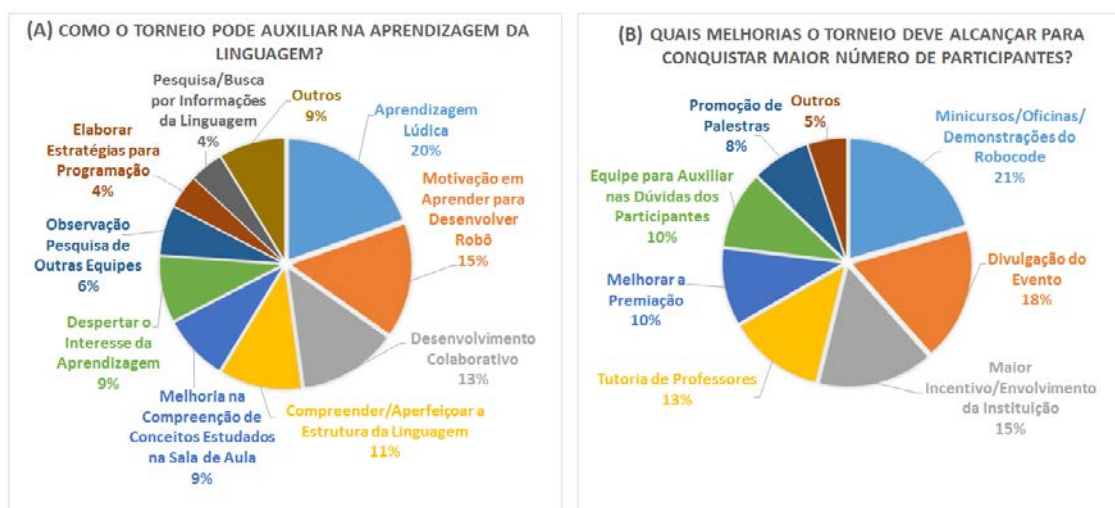


Figura 3 - (a) Torneio na Aprendizagem (b) Conquista de Participantes.

A realização dos torneios em caráter interinstitucional valorizou aspectos de colaboração entre instituições, pôde exibir que ao ser incentivado, estes eventos podem

resultar na diferenciação de metodologias de aprendizagem. Esse fato ficou evidente na equipe campeã do torneio de 2015, IF Farroupilha, que após a conquista do título gravou um vídeo e publicou nota em agradecimento e incentivo as iniciativas Robocode. Esse resultado obteve repercussão positiva interinstitucional apresentando motivações de novos alunos e instituições a conhecer os torneios Robocode.

Na Figura 4 (a), observa-se a aplicação de conceitos presentes nos planos de ensino da disciplina de linguagem de programação, plano de aula de conteúdo programático referente aos “comandos de decisões” *if-else* (se-senão). A Figura 4 (a) exibe o código de uma equipe do curso técnico em informática com noções ou conhecimento superficial da linguagem de programação Java. Essa equipe propôs “variáveis booleanas” ($\text{ângulo} == \text{true}$) com instruções para “tomada de decisão” (*if-else*) quanto à movimentação do robô. O exemplo indica *if* (se) o ângulo é verdadeiro, o robô deve avançar o valor de 100 e virar à direita em 90 graus, *else* (senão) ângulo é falso e o robô deve recuar o valor de 50 e virar para esquerda em 90 graus.

<pre> 5 public void run() { 6 boolean angulo = true; 7 while(true) { 8 if (angulo == true){ 9 ahead(100); turnRight(90); 10 angulo = false;} 11 else{ 12 back(50); turnLeft(90); 13 angulo = true;} 14 } } </pre>	(A)	<pre> 16 public void run() { 17 while(true) { 18 for (int virarDir=0;virarDir<4;virarDir++){ 19 ahead(100); 20 turnRight(90);} 21 for (int virarEsq=4;virarEsq>0;virarEsq--){ 22 ahead(100); 23 turnLeft(90);} 24 } 25 } </pre>	(B)
--	------------	---	------------

Figura 4 – (a) Comandos de Decisão (b) Estruturas de Repetição

Os códigos exibidos na Figura 4 (a) e (b) foram desenvolvidos pela mesma equipe em momentos distintos dentro da competição. Código (b) foi criado em rodadas posteriores ao (a). A Figura 4 (b) indica o uso do conceito de “estrutura de repetição” com comando *for* (para), apresentado na linha 18. Na mesma linha, observa-se o uso do comando *for* com sua “variável de inicialização” (int), as “condicionais” recebendo valores (inicial: $\text{virarDir}=0$; final: $\text{virarDir}<4$) e ao final seu “incremento” ($\text{virarDir}++$). Para entender o conceito a equipe pesquisou informações dos conceitos envolvidos acerca da “variável de inicialização” (inicia um controle de repetição), as “condições” (determina a condição inicial e final da repetição) e “incremento” (variável de controle alterada a cada repetição). Após o entendimento dos conceitos, a equipe pôde aferir o resultado em comandos de movimentação do robô: avançar no valor de 100 e virar à direita em ângulo de 90 graus por 4 vezes consecutivas; em seguida repetir o mesmo processo à esquerda.

5. Conclusão

Alunos das gerações atuais, inseridos em um contexto de conectividade no universo digital, estão sujeitos a encontrar ambientes de ensino tradicionais desmotivadores. Estes ambientes podem desestimular alunos acostumados a interagir com tecnologias digitais. Este trabalho apresentou experiências com a organização de competições científicas baseadas em Robocode. Foi apresentada uma experiência de condução de competições desde 2010.

As competições científicas que envolvem torneios e ligas do jogo educacional Robocode têm funcionado como promissoras ferramentas de diversificação de métodos de aprendizagem nas instituições que a realizam. Além de introduzir métodos diferenciados, as competições estimulam o aprendizado lúdico e incorporam aspectos do aprendizado colaborativo.

Com base nas experiências apresentadas, os torneios Robocode podem motivar participantes a buscar informações para estudo e desenvolvimento dos conceitos da linguagem de programação em ambientes coletivos. Nestas competições, os problemas apresentados são dinâmicos e geram estímulos para que as equipes tenham condições de trabalhar no desenvolvimento das soluções de acordo com situações visualizadas durante a execução das partidas.

As regras do torneio indicam que a cada nova rodada os códigos devem ser abertos. Essa característica garante as equipes com noções ou conhecimentos superficiais tenham a possibilidade de analisar códigos elaborados por outras equipes de alunos com prévio entendimento da linguagem. A partir de análises mútuas do desenvolvimento de códigos, equipes buscam entender e estudar novos conceitos com objetivo de aperfeiçoar as características dos robôs. A proposta com ciclo de abertura dos códigos fonte dos robôs durante o torneio favorece o nivelamento para estudos de conceitos como “comandos de decisões” ou “estruturas de repetição” discutidos no presente artigo. Existem ainda as discussões de estratégias de jogo e compartilhamento de experiência entre equipes ao longo da execução do torneio.

Para vencer o torneio, as equipes devem trabalhar o desenvolvimento contínuo de estratégias para suportar as diversas situações exigidas no jogo, como a melhoria da movimentação dos robôs. Em estágios avançados do torneio, equipes com movimentação na base do conceito de “comando de decisão” sentiram a necessidade de explorar novos conceitos ainda não estudados, como as “estruturas de repetição”. Observou-se a evolução das equipes durante o torneio, com o estudo de conceitos de linguagem de programação presentes nos planos de ensino dos cursos muitas vezes ainda não tratados nas disciplinas tradicionais. Torneios com envolvimento de jogos digitais, como Robocode, podem auxiliar na diversificação da aprendizagem. Os ambientes dos torneios geram situações-problemas que podem incentivar alunos a buscar soluções, na aprendizagem de novos conceitos, como aperfeiçoamento de estratégias de movimentação dos robôs. Equipes formadas por alunos de diferentes níveis de conhecimento, tiveram a oportunidade de estabelecer um nivelamento de conceitos da linguagem. Resultados positivos, enfatizadas pelas instituições e entrevistados, indicaram o jogo Robocode associado aos torneios podem auxiliar no estímulo do processo de aprendizagem de linguagem de programação de modo lúdico.

7. Referências Bibliográficas

- Adams, D. M.; Mayer, R. E.; Koenig, A. W. (2012) “Narrative Games for Learning: Testing the discovery and narrative hypotheses”. *Journal of Educational Psychology*, 104 (1), pp. 235-249, doi: 10.1037/a0025595.
- Barnes, T.; Richter, H.; Powell, E.; Chaffin, A.; Godwin, A. (2007) “Game2Learn: Building CS1 Learning Games for Retention”. *SIGCSE Bull.* 39, 3 (June 2007), 121-125, doi=<http://dx.doi.org/10.1145/1269900.1268821>.

- Chaffin, A.; Doran, K.; Hicks, D.; Barnes, T. (2009) “Experimental Evaluation of Teaching Recursion in a video game”. In Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games (Sandbox '09), Stephen N. Spencer (Ed.). ACM, New York, NY, USA, 79-86, doi=<http://dx.doi.org/10.1145/1581073.1581086>.
- Fayek, M. B.; Farag, O. S. Hicma. (2014) “A Human Imitating Cognitive Modeling Agent Using Statistical Methods and Evolutionary Computation”. In Computational Intelligence for Human-like Intelligence (CIHLI), 2014 IEEE Symposium on, vol., no., pp.1-8, 9-12, doi: 10.1109/CIHLI.2014.7013383, Dec.
- Gallant, R. J.; Mahmoud, (2008) “Q. H. Using Greenfoot and a Moon Scenario to teach Java programming in CS1”. Proceedings of the 46th Annual Southeast Regional Conference on XX - ACM-SE 46 (p. 118). New York, New York, USA: ACM Press. doi: 10.1145/1593105.1593135.
- Garcia. R. E.; Correia, R. C. M.; Shimabukuro. M. H. (2008) Ensino de Lógica de Programação e Estruturas de Dados para Alunos do Ensino Médio. Anais XXVIII Congresso SBC. WEI–Workshop sobre Educação em Computação, Belém-PA, julho.
- Hong, J. H.; Cho, S. B. (2004) “Evolution of Emergent behaviors for shooting game characters in Robocode”. In Evolutionary Computation, 2004. CEC2004. Congress on, vol.1, no., pp.634-638 Vol.1, 19-23, doi: 10.1109/CEC.2004.1330917, June.
- Hou, S. I. (2014) “Integrating Problem-based Learning With Community-Engaged Learning in Teaching Program Development and Implementation”. Universal Journal of Educational Research 2(1): 1-9, doi: 10.13189/ujer.2014.020101.
- Johnson, W. L. (2010) “Serious Use of a Serious Game for Language Learning”. International Journal of Artificial Intelligence in Education, 20 (2), pp. 175-195, doi: 10.3233/JAI-2010-0006.
- Li, S. (2002) “Rock 'em, sock 'em Robocode!” Learning Java programming is more fun than ever with this advanced robot battle simulation engine. IBM developerWorks, <http://www.ibm.com/developerworks/java/library/j-robocode/>.
- Mezzari, A. (2011) O Uso da Aprendizagem Baseada em Problemas (ABP) como reforço ao ensino presencial utilizando o ambiente de aprendizagem Moodle. Revista Brasileira de Educação Médica. Rio de Janeiro. v. 35, n. 1, p. 114-121, jan./mar.
- Nidorf, D.G.; Barone, L.; French, T. (2010) “A comparative study of NEAT and XCS in Robocode”. In Evolutionary Computation (CEC), 2010 IEEE Congress on , vol., no., pp.1-8, 18-23, doi: 10.1109/CEC.2010.5586087, July.
- Prensky, M. (2012) Aprendizagem Baseada em Jogos. Tradução de Eric Yamagute; revisão técnica de Romero Tori e Denio di Lascio. Editora Senac, São Paulo.
- Recchia, T., Chung, J., and Pochiraju, K. (2014). “Performance of Heterogeneous Robot Teams With Personality Adjusted Learning”. Biologically Inspired Cognitive Architectures, vol. 7, pp. 87–97, doi=<http://dx.doi.org/10.1016/j.bica.2013.10.003>.
- Robocode. (2013) “Open Source Educational Game, ReadMe for Robocode”. Disponível em: <<http://robocode.sourceforge.net>>. Acesso em: agosto de 2015.
- Valente, J. A. (1999) O Computador na Sociedade do Conhecimento. São Paulo: UNICAMP/NIED.