

Aprendizagem de Linguagem de Programação com Metodologia PBL em Competições Científicas com Robocode

Matheus C. Meira¹, 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, marcosborges@ft.unicamp.br

Abstract. *Students of younger generations, inserted in a context of connectivity in the digital universe may consider demotivating the traditional teaching environment for programming disciplines. This study presents an approach to teaching-learning programming language, with Problem-Based Learning, in the development of scientific competitions based on the digital educational game Robocode. The objective is associated with evaluating whether scientific competitions with Robocode stimulate learning of computer programming concepts in technical and superior courses of Computer Science area. The approaches of the study point to complementary actions in programming language disciplines.*

Resumo. *Alunos das gerações atuais, inseridos em um contexto de conectividade no universo digital, podem considerar desmotivadores ambientes de ensino tradicionais para disciplinas de programação. Este trabalho apresenta uma abordagem para ensino-aprendizagem de linguagem de programação, com Aprendizagem Baseada em Problemas, no desenvolvimento de competições científicas baseadas no jogo digital educacional Robocode. O objetivo está associado em avaliar se competições científicas com Robocode estimulam aprendizagem de conceitos de programação de computadores em cursos técnicos e superiores da área de informática. As abordagens do estudo apontam ações complementares em disciplinas de linguagem de programação.*

1. Introdução

Atualmente, instituições de ensino desenvolvem ações com jogos educacionais digitais nos processos de ensino-aprendizagem para incentivar o aprendizado de programação [Barnes et al. 2007; Chaffin et al. 2009]. Conteúdos de programação em si não caracterizam apenas assuntos exclusivos das disciplinas específicas de linguagem de programação. Esses 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, envolvidos em tecnologias cada vez mais presentes em suas vidas. A programação consegue despertar a criatividade das pessoas que a utilizam [Valente 1999]. Alunos participantes de disciplinas de programação ou admiradores de tecnologias, muitas vezes têm o anseio de entender a lógica utilizada nos aplicativos atuais. Este trabalho busca apresentar e discutir abordagens interessantes para incentivar a aprendizagem de programação entre alunos de cursos técnicos e superiores da área de informática.

“O método tradicional de ensino é centrado no professor, que possui o papel de sujeito ativo e o aluno passivo no processo de aprendizagem” [Mezzari 2011]. A ênfase do ensino tradicional está na transmissão dos conhecimentos pelo professor, que domina e organiza os conteúdos a serem repassados aos alunos. Com papel de ouvintes, os alunos

têm como principal função memorizar informações transmitidas pelo professor [Saviani 2012]. Mesmo em disciplinas de linguagem de programação, no instante em que o professor adota a abordagem do computador como meio para transmitir informação ao aluno, “a máquina está sendo usada para informatizar processos de ensino existentes. Isso facilita a implantação e o uso do computador nas instituições de ensino, porém não quebra a dinâmica do método tradicional” [Valente 1999].

O Aprendizado Baseado em Problemas (*Problem Based Learning* - 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 presente trabalho exercitou uma dinâmica de aprendizagem a partir da associação da metodologia PBL com jogos educacionais digitais em competições científicas¹ para propor e analisar um ambiente de aprendizagem de linguagem de programação. O software, ambiente do jogo educacional Robocode, compõe o cenário de aplicação da metodologia PBL. Robocode consiste em um jogo de programação, composto por um ambiente de desenvolvimento, que permite ensinar linguagem de programação (Java e C#) de modo lúdico [Bonakdarian and White 2004].

A pesquisa correlata realizada por Piekarski et al. (2015) apresentou a experiência de um projeto de extensão, a partir de conceitos que envolvem competições de programação. A pesquisa teve por objetivo o treinamento de alunos em linguagem de programação para disputa de maratonas de programação. Essa experiência visou a seleção de equipes para participar da primeira fase de umas das maratonas de programação mais importantes realizadas no Brasil (Maratona de Programação Sociedade Brasileira de Computação – SBC) [Piekarski et al. 2015].

Neste contexto, faz-se a seguinte questão: a combinação da metodologia PBL com jogo educacional Robocode em competições científicas pode auxiliar na motivação dos alunos durante o processo de aprendizagem em disciplinas que envolvem linguagem de programação? Hipóteses para responder a esse questionamento remetem na organização de um ambiente propício a competição científica de programação que trabalhe na geração de desafios e que despertem o desenvolvimento de pesquisas e trabalhos colaborativos. Um ambiente para incentivar o despertar do interesse no aprendizado com base na linguagem de programação Java, a partir de elementos que facilitam o uso de determinadas estruturas lógicas dispostas no jogo educacional Robocode.

O objetivo principal deste trabalho é avaliar se competições científicas com base nos conceitos de linguagem de programação Java presentes no Robocode e estruturadas no PBL tem potencial para estimular a aprendizagem em disciplinas que envolvam linguagem de programação.

A crescente retenção dos alunos do primeiro ano dos cursos técnicos, integrados e superiores na área de informática retrata uma preocupação das instituições participantes da competição científica. É possível citar os campi do Instituto Federal de Educação Ciência e Tecnologia de São Paulo (IFSP), da qual apontam a falta de ações pedagógicas em disciplinas relacionadas a programação de computadores, que apresentam altas taxas

¹ Competições Científicas e Olimpíadas Científicas recebem definições similares de acordo com ICMC da USP e CNPq: momentos privilegiados para a divulgação científica e para a descoberta e incentivo de novos talentos. O caráter competitivo estimula a inventividade dos alunos e professores”; “buscam incentivar o trabalho em equipe, reforçando hábitos de estudo, o despertar de vocações científicas e os vínculos de cooperação entre equipes de estudantes e professores.

de retenção. A retenção por conteúdo ocorre quando alunos apresentam dificuldades de assimilação e pode provocar a evasão escolar [FEPESP 2011].

Não se pode fixar apenas em meios tradicionais de ensino. É importante combinar diferentes métodos para atrair e incentivar os alunos a conquistar os objetivos propostos no currículo da disciplina. As instituições de ensino podem obter benefícios com os avanços tecnológicos dos softwares educacionais para auxiliar o processo de ensino-aprendizagem. Para isso, podem utilizar softwares educacionais, do tipo jogos, combinados com métodos para desenvolver ambientes de aprendizagem em disciplinas de linguagem de programação.

2. Framework de Simulação “Robocode”

O projeto Robocode é um software Open Source, com fontes abertas e sob os termos da EPL (*Eclipse Public License*), que possibilita simulações de disputas com robôs virtuais com base nas linguagens de programação Java e C# [Fayek and Farag 2015; Robocode 2013]. O Robocode foi projetado para ser um ambiente de simulação de robôs em uma arena, com o objetivo de comparar estratégias de programação com disputas estruturadas, tendo cada robô seus próprios algoritmos e comportamentos.

O jogo Robocode permite o ensino de linguagem de programação Java ou C#. O objetivo da presente abordagem está centrado na apresentação do jogo educacional Robocode em competições para estímulo de aprendizagem de linguagem de programação. O Robocode pode ser explorado como plataforma no desenvolvimento de estratégias de jogo a partir de competições científicas planejadas em um ambiente de ensino-aprendizagem de programação de computadores.

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. No Robocode, o agente tem a possibilidade de escolher uma variedade de comportamentos e medir a eficácia de cada um ao longo do desafio [Nidorf et al. 2010].

Robocode pode ser enquadrado como um *framework* de simulação de fácil utilização, originalmente criado para ensinar Programação Orientada à Objetos (POO) [Alaiba and Rotaru 2008]. A classificação do Robocode está centrada em um simulador de batalhas com base em Inteligência Artificial (IA) [Recchia et al. 2014]. No simulador de disputas, é possível observar o comportamento dos robôs. Pelo fato das disputas 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.) [Fayek and Farag 2015]. A Figura 1 apresenta visões acerca de enquadramentos para diferentes características do Robocode.



Figura 1 - Definições para Robocode.

3. Ambiente de Competição com Robocode

Competições científicas de programação podem oferecer abordagens motivadoras aos alunos para introduzir conceitos de programação e encontrar novos talentos. Motivar alunos a aprender consiste em um importante tema de discussão entre professores. Neste

contexto, as competições apresentam-se como abordagens que podem oferecer ampla gama de possibilidades para auxiliar e envolver alunos no aprendizado de conceitos de programação [Hakulinen 2011]. Podem ser considerados objetivos das competições científicas: aproximar uma disciplina específica do cotidiano dos alunos; apresentar aplicações práticas na resolução de problemas reais ou simulados; e envolver de modo natural o interesse dos alunos. As competições também despertam novos vínculos dos alunos perante a escola e incentivam trabalhos em equipes com definições de estratégias cooperativas de aprendizagem. Uma competição trabalha valores afetivos como a autoconfiança e autoestima do aluno, na medida da evolução das disputas, associando-as a sua capacidade de interagir no grupo e resolver problemas. Atualmente, as competições também são empregadas de acordo com Science Olympiad (2016) para atrair estudantes do sexo feminino nas carreiras de áreas como ciência e tecnologia [ScienceOlympiad 2016].

Em relação as competições do 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 Robowiki² associada a iniciativa colaborativa denominada *RoboRumble*³ 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 (nível médio e superior) 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.

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. Universidades Federais espalhadas e Instituições Privadas em todos Brasil como: UFU - Universidade Federal de Uberlândia; UFC - Universidade Federal do Ceará; Unisal; Anhangüera; PUC; dentre outras, têm vasta tradição na execução de torneios ao longo dos anos.

² Robowiki. (2016) *Collecting Robocode Knowledge Since 2003*. Disponível em: <<http://robowiki.net>>. Código fonte aberto, com a promoção de competições ao redor do mundo.

³ RoboRumble (2016) Disponível em: <<http://robowiki.net/wiki/RoboRumble>>. Esforço colaborativo mantido pela RoboWiki com objetivo de gerenciar e manter rankings *on-line* atualizados das disputas mundiais de robôs distribuídas em categorias do Robocode.

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

3.1. Robocode Brasil 2016

O torneio Robocode do Laboratório de Informática, Aprendizado e Gestão (LIAG⁵) da UNICAMP ocorre desde de 2010. Os torneios Robocode LIAG são regidos por regras que visam 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 a Liga. 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.

Este trabalho buscou analisar a utilização do método de PBL em competições científicas com jogo educacional Robocode para motivar alunos no estudo da disciplina de linguagem de programação. PBL representa um método de ensino-aprendizagem centrado no aluno, com estratégias voltadas à resolução de problemas. O problema é o *design* de um robô, estando alinhando com as regras do jogo e da competição Robocode.

O estudo de caso deste trabalho foi baseado em ação em parceria do LIAG da Faculdade de Tecnologia (FT) da Universidade Estadual de Campinas (UNICAMP) com o Instituto Federal de Educação Ciência e Tecnologia (IFSP⁶). As instituições de ensino que compuseram o ambiente da competição científica foram: FT-UNICAMP – Faculdade de Tecnologia (Limeira - SP); COTIL-UNICAMP - Colégio Técnico de Limeira (Limeira - SP); IFSP – Instituto Federal São Paulo (Capivari – SP); IFSP – Instituto Federal São Paulo (Hortolândia – SP); IF Farroupilha - Instituto Federal Farroupilha (Campus São Borja - RS); UNICAMP - Universidade Estadual de Campinas (Campinas – SP); COTUCA - Colégio Técnico de Campinas (Campinas - SP); ETEC - Escola Técnica Estadual Adolpho Berezin (Mongaguá – SP); Faculdade Anhanguera (Rio Claro - SP); Faculdade Anhanguera (Santa Bárbara d'Oeste – SP); Faculdade SATC (Criciúma – SC); PUC - Pontifícia Universidade Católica do Paraná (Londrina – PR); UFBA - Universidade Federal da Bahia (Salvador – BA); UFMG - Universidade Federal de Minas Gerais (Belo Horizonte – MG). Os dados das competições científicas avaliados no trabalho são referentes ao ano de 2015 e 2016. A competição científica proposta no ano de 2016 recebeu o nome de Robocode Brasil⁷.

Em comparação ao ano de 2015, a Figura 2 exibe o Robocode Brasil 2016. Observa-se no mapa o aumento na quantidade de instituições, passando de 5 (cinco) para um total de 14 (quatorze) instituições participantes, envolvendo cerca de 170 (cento e setenta) alunos inscritos e 50 (cinquenta) equipes. O Robocode Brasil 2016 consistiu na maior competição brasileira com ambiente Robocode, ultrapassando sua antecessora (Liga

⁵ Laboratório da FT-UNICAMP com a missão de integrar os estudos nas áreas de informática, aprendizagem e gestão de modo a buscar resultados superiores e diferenciados nas três áreas.

⁶ Autarquia Federal de ensino vinculada ao Ministério da Educação (MEC) fundada em 1909 e reconhecida pela sociedade paulista por sua excelência no ensino público gratuito de qualidade

⁷ Robocode Brasil, <http://www.robocodebrasil.com.br>, consiste no canal oficial da liga brasileira de Robocode. O ambiente gerenciador da competição permite: inscrições e agrupamentos das instituições; equipes e participantes; receber códigos e desempenhar as partidas; gerenciar os resultados e produzir relatórios de status; consulta as regras e edital da competição; canal de comunicação com *newsletter* e capacitações no Robocode.

LIAG Robocode 2015). Apresentou 7 (sete) Torneios Locais (instituições na cor rosa na Figura 2) e 1 (um) Torneio Público (instituições na cor verde na Figura 2).

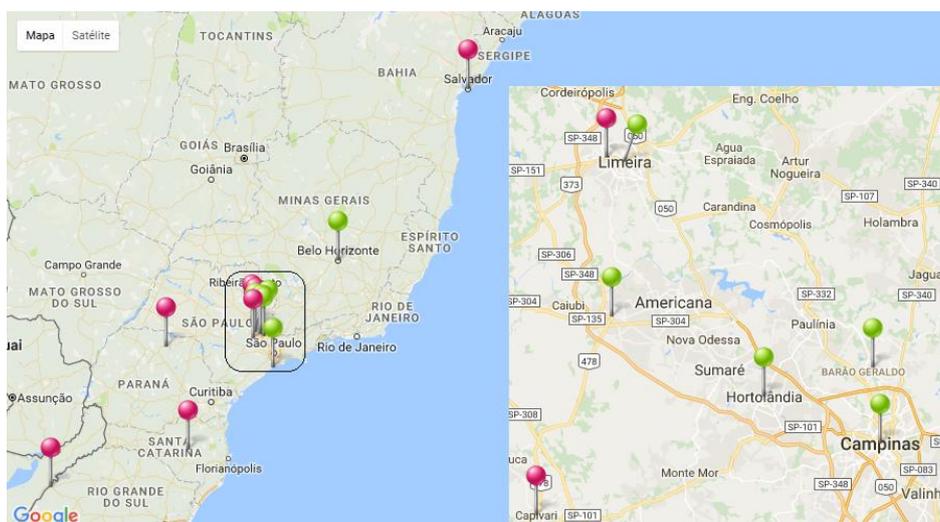


Figura 2 - Mapa Referenciado das Instituições do Robocode Brasil 2016.

4. Discussão

Um dos objetivos deste trabalho foi avaliar se competições científicas com Robocode estimularam aprendizagem de conceitos de linguagem de programação em cursos técnicos e superiores da área de informática. Para isso, foram desenvolvidos 2 (dois) questionários para analisar a experiência dos alunos em dois momentos distintos da competição Robocode Brasil (2015 e 2016). Os questionários foram denominados de inicial e final e foram respondidos individualmente por cerca de 140 alunos.

Uma das perguntas (Q1) relaciona a incidência de cursos, nível e tipo de curso das quais participam os alunos inscritos na competição. A Figura 3 (a) mostra os cursos dos alunos inscritos na competição. Mais de 50% dos alunos participantes da competição foram dos cursos Técnicos Integrados em Informática e cursos superiores de Bacharelado em Ciência da Computação somados, indicados na Figura 3 (a) nas cores laranja (27,7%) e azul (27,7%). É possível observar a incidência do curso de Tecnologia em Jogos Digitais dos alunos da PUC de Londrina-PR. A Figura 3 (b) apresenta o nível do curso, com 57,4% de nível superior e 42,6 de nível técnico/técnico integrado. A Figura 3 (c) exibe o tipo de curso com destaques para Bacharelado, 40,4% (Ciência da Computação e Sistemas de Informação), e Técnico Integrado, 24,5% (Técnico Integrado em Informática).

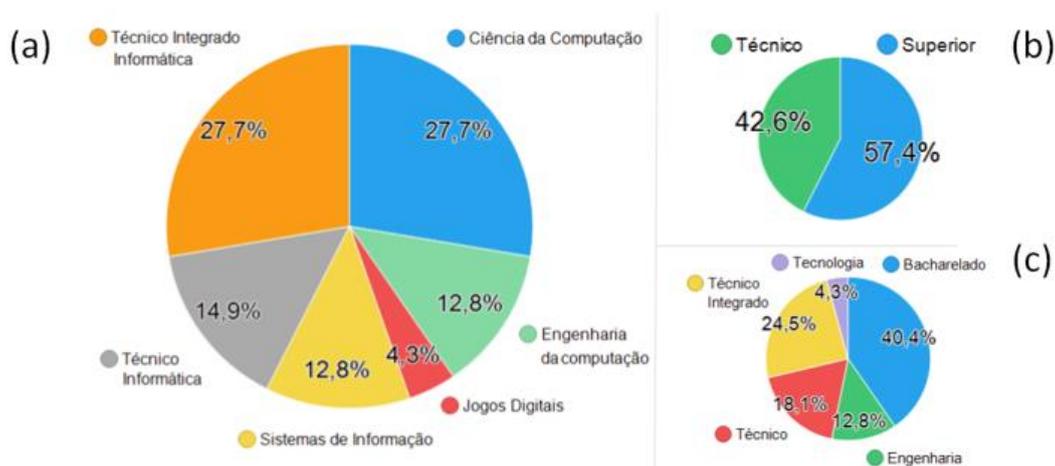


Figura 3 – “Q1” (a) Curso; (b) Nível; (c) Tipo de Curso.

Para a questão “Q2” fez-se importante ressaltar que os processos da metodologia PBL admitem etapas e processos distintos para determinação dos problemas. Os processos PBL foram utilizados para determinar os problemas gerados automaticamente durante as disputas na competição Robocode Brasil 2016. A Tabela 1 exibe os principais processos do PBL e sua abordagem dentro da competição Robocode Brasil.

Tabela 1 – Processos do PBL Associados à Competição

Etapa	Processos Metodologia PBL	Abordagem PBL na Competição
1	1. Identificação e delimitação do problema	1. Analisar os problemas e as estratégias suportadas na competição e no jogo.
	2. Formular os objetivos de aprendizagem a partir da discussão do problema.	2. Formular objetivos de acordo com os métodos do robô (ex. ajustar a movimentação do robô).
2	3. Levantar as hipóteses possíveis para resolução do problema	3. Promover debates e discussões na equipe para resolver os problemas.
	4. Rediscussão do problema com base nos novos conhecimentos adquiridos	4. Rediscutir os problemas com base nos novos conceitos de linguagem de programação e estratégias aprendidos.

A questão “Q3” teve por objetivo principal analisar se os 4 (quatro) principais processos do PBL foram atendidos durante a resolução dos problemas propostos na abordagem da competição. A Figura 4 apresentou os resultados, nos quais é possível separar os termos “sempre” e “frequentemente”, amplamente citados e reiterados na ordem dos processos e abordagens da Tabela 1: (1) 65,4% – identificação e estratégias; (2) 72,3% – formulação de objetivos; (3) 58,3% – discussões em grupo; (4) – 68,1% rediscussão dos problemas. Para a utilização de modo “moderado” o resultado estabeleceu-se: (1) 27,8% – identificação e estratégias; (2) 19,4% – formulação de objetivos; (3) 26,4% – discussões em grupo; (4) – 22,3% rediscussão dos problemas. Para “raramente” ou “nunca” houve baixos níveis de incidências: (1) 6,8% – identificação e estratégias; (2) 8,3% – formulação de objetivos; (3) 15,3% – discussões em grupo; (4) – 9,6% rediscussão dos problemas.

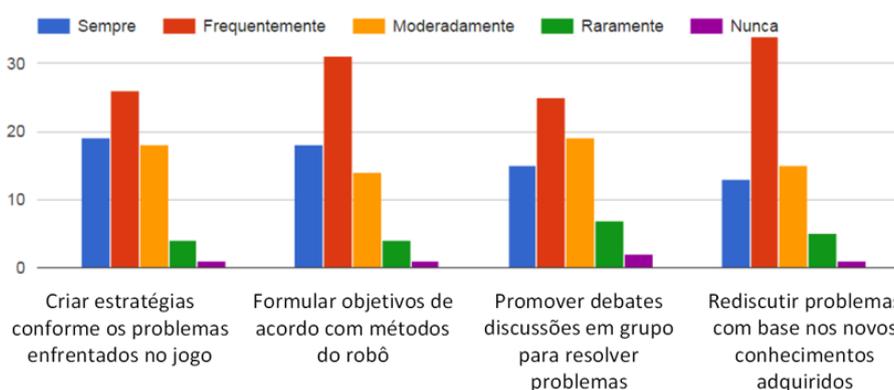


Figura 4 – “Q3” Utilização dos Processos PBL na Competição.

Em instituições de ensino que oferecem cursos técnicos ou superiores na área de informática, é comum para alunos ingressantes o primeiro contato com disciplinas que envolvam programação de computadores logo nos primeiros meses. Professores que ministram estas disciplinas devem, antecipadamente, preparar planos de ensino e aulas a fim de distribuir e balancear corretamente os conteúdos, conforme a carga horária da disciplina. Conteúdos introdutórios ou conceitos de programação, em geral, são apresentados logo no início das disciplinas relacionadas à programação. Os problemas apresentados às equipes, durante a competição, tiveram por objetivos atender conceitos introdutórios da linguagem de programação presentes nos planos de ensino/aula. Na

Figura 5 (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 5 (a) exibe o código de uma equipe do curso técnico integrado 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 [Meira et al. 2016].

<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>	<p>(A)</p>	<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>	<p>(B)</p>
--	-------------------	---	-------------------

Figura 5 – (a) Comandos de Decisão (b) Estruturas de Repetição

Os códigos exibidos na Figura 5 (a) e (b) foram desenvolvidos pela mesma equipe em momentos distintos dentro da competição. O Código (b) foi criado em rodadas posteriores ao (a). A Figura 5 (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 [Meira et al. 2016].

5. Conclusão

Alunos de gerações atuais inseridos em um contexto de conectividade no universo digital tendem a considerar ambientes de ensino tradicionais desmotivadores. Essa desmotivação pode interferir na estruturação do conteúdo aprendido, enfatizando um conhecimento apenas superficial. Este trabalho propôs uma nova abordagem para ensino-aprendizagem, com geração de interações, apropriando-se de ferramentas presentes no cotidiano dos alunos para ensino de linguagem de programação.

Um ambiente de competição científica de programação trabalha com diversas situações problemas que permitem desenvolver a colaboração. A estruturação da metodologia PBL na competição científica encoraja não somente um conhecimento mais profundo de uma determinada linguagem de programação, Java, mas também de estruturas lógicas que de fato resolvem um problema. O jogo educacional Robocode pode despertar o interesse no aprendizado de conceitos de linguagem de programação a partir de elementos que facilitam o uso de determinadas estruturas lógicas.

Dentro das instituições participantes, 80,9% dos alunos nunca participaram de outras iniciativas com jogos, tendo sido a primeira experiência da maioria. Entrevistas

com organizadores de cada uma das competições revelaram a baixa incidência de iniciativas. Para *Serious Games* de programação, 83% desconheciam qualquer jogo ou ambiente com essa finalidade. Ficou evidenciado que pode ser interessante que as instituições desenvolvam iniciativas nas temáticas dos jogos digitais na área de informática. Para o Robocode Brasil 2016, 89,7% dos alunos admitiram auxílio da competição no entendimento de conceitos. Outros 97,1% consideraram os jogos, ênfase no Robocode, como ferramentas de auxílio a serem utilizadas nas aulas. Os resultados apontaram o interesse dos alunos no ensino-aprendizagem com novas propostas diversificadas com *Serious Games*.

Para vencer a competição, 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 a competição, com o estudo de conceitos de linguagem de programação presentes nos planos de ensino dos cursos muitas vezes ainda não tratados nas disciplinas tradicionais. Competições com envolvimento de jogos digitais, como Robocode, podem auxiliar na diversificação da aprendizagem. Os ambientes das competições 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, enfatizados pelas instituições e entrevistados, indicaram o jogo Robocode associado as competições podem auxiliar no estímulo do processo de aprendizagem de linguagem de programação de modo lúdico.

Ao final, pode-se observar que o desenvolvimento de métodos diversificados em competições científicas com jogo educacional Robocode pode ser uma abordagem efetiva quando o objetivo é auxiliar a estimular alunos na aprendizagem de conceitos de linguagem de programação. O trabalho realizado com Robocode Brasil 2016 concluiu na satisfação geral com cerca de 75% dos alunos, participantes dos questionários, estiveram satisfeitos ou muito satisfeitos com os resultados da competição. Essa alta incidência de porcentagem de satisfação, também, pode ser caracterizada como estímulo ao aprendizado. O Robocode Brasil 2016 serviu como ação complementar em disciplinas de programação nos cursos da área de informática.

Trabalhos futuros remetem a adoção de contextos de forma criativa como tecnologias emergentes da Aprendizagem Baseada em Jogos, com comunidades de aprendizagem, a partir de projetos que envolvam competições científicas ou oficinas que extrapolem o limite da escola. Avaliar conceitos, práticas desenvolvidos e, principalmente, estudar a influência dos jogos digitais nessas comunidades de aprendizagem.

7. Referências Bibliográficas

- Alaiba, V. and Rotaru, A. (2008). Agent architecture for building robocode players with SWI-Prolog. *Proceedings of the International Multiconference on Computer Science and Information Technology, IMCSIT 2008*, v. 3, n. 2, p. 3–7.
- Barnes, T., Powell, E., Chaffin, A., Goldwin, A. and Richter, H. (2007). Game2Learn : Building CS1 Learning Games for Retention. *ACM SIGCSE Bulletin*, v. 39, n. 3, p. 121–125.

- Bonakdarian, E. and White, L. (2004). Robocode Throughout the Curriculum. *Journal of Computing Sciences in Colleges - JCSC. Consortium for Computing Sciences in Colleges - CCSC: Southeastern Conference.*, v. 19, n. 3, p. 311–313.
- Chaffin, A., Doran, K., Hicks, D. and Barnes, T. (2009). Experimental evaluation of teaching recursion in a video game. *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games - Sandbox '09*, v. 1, n. 212, p. 79.
- Fayek, M. B. and Farag, O. S. (2015). HICMA: A human imitating cognitive modeling agent using statistical methods and evolutionary computation. *IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - CIHLI 2014: 2014 IEEE Symposium on Computational Intelligence for Human-Like Intelligence, Proceedings*,
- FEPEESP (2011). Evasão na Educação Profissional. *Fórum da Educação Profissional do Estado de São Paulo - FEPEESP, 31 de maio de 2011, São Paulo. Anais eletrônicos.*, p. 1–1.
- Hakulinen, L. (2011). Survey on Informatics Competitions: Developing Tasks. *Olympiads in Informatics*, v. 5, p. 12–25.
- Hou, S.-I. (2014). Integrating Problem-Based Learning with Community-Engaged Learning in Teaching Program Development and Implementation. *Universal Journal of Educational Research*, v. 2, n. 1, p. 1–9.
- Li, S. (2002). Rock 'em, sock 'em Robocode!, Learning Java programming is more fun than ever with this advanced robot battle simulation engine. <http://www.ibm.com/developerworks/java/library/j-robocode>, [accessed on Jan 1].
- Meira, M. C., Lima, M. S. S. and Borges, M. A. F. (2016). Torneios Baseados em Robocode para Incentivar Jovens a Aprender Programação. *Anais dos Workshops do XXXVI Congresso da Sociedade Brasileira de Computação (CSBC 2016)*, n. CSBC, p. 2403–2412.
- 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*, v. 35, n. 1, p. 114–121.
- Nidorf, D. G., Barone, L. and French, T. (2010). A comparative study of NEAT and XCS in robocode. *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, p. 1–8.
- Piekarski, A. E., Miazaki, M., Hild, T., Mulati, M. H. and Kikuti, D. (2015). A metodologia das maratonas de programação em um projeto de extensão: um relato de experiência. n. Cbie, p. 1246.
- Recchia, T., Chung, J. and Pochiraju, K. (jan 2014). Performance of heterogeneous robot teams with personality adjusted learning. *Biologically Inspired Cognitive Architectures*, v. 7, p. 87–97.
- Robocode (2013). Open Source Educational Game, ReadMe for Robocode. <http://robocode.sourceforge.net>, [accessed on Jun 1].
- Saviani, D. (2012). *Escola e Democracia*. 42. ed. Campinas: Autores Associados.
- ScienceOlympiad (2016). Science Olympiad. <https://www.soinc.org>, [accessed on Oct 10].
- Valente, J. A. (1999). *O Computador na Sociedade do Conhecimento*. Campinas, São Paulo: OEA_NIED/UNICAMP.