

CodeBots

Ensino Lúdico de Conceitos Introdutórios de Programação para Estudantes da Educação Básica

Felipe Oviedo Frosi
Faculdade de Informática
Centro Universitário Ritter dos Reis (UniRitter)
Porto Alegre, Brasil
Email: felipefrosi@hotmail.com

Isabel Cristina Siqueira da Silva
Programa de Pesquisa e Pós-Graduação em Arquitetura
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brasil
Email: isabel.siqueira@gmail.com

Resumo—O aprendizado de lógica de programação, desde a formação básica, ajuda os estudantes a desenvolver habilidades para resolver problemas, incentiva o trabalho em equipe e aumenta a capacidade destes de pensar de forma sistematizada e criativa. Nota-se, então, a constante necessidade pela proposição de ferramentas que auxiliem o aprendizado em Computação de forma lúdica e interativa. Neste contexto, este artigo apresenta os resultados de uma pesquisa que teve como objetivo analisar a utilização do jogo digital CodeBots no contexto de uma escola estadual de ensino fundamental. A análise foi realizada através da utilização do jogo seguida da realização de um teste escrito envolvendo conceitos introdutórios de programação. Como resultado são elencados aspectos sobre como o uso da tecnologia pode contribuir para o ensino e a aprendizagem das noções de programação no contexto escolar, além de reflexões sobre possíveis aprimoramentos para futuras aplicações do jogo e da importância de manter a tecnologia atualizada para engajar os estudantes.

Keywords—jogos digitais; programação; jogos educacionais; ludicidade.

I. INTRODUÇÃO

Conceitos de lógica de programação estão presentes no cotidiano das pessoas, seja na sequência de ações para preparar o café da manhã, para se deslocar até a escola e/ou trabalho, para comprar artigos em um supermercado, entre outros. No entanto, a conversão desses “algoritmos do dia a dia” para o aprendizado formal de lógica de programação, geralmente, não é trivial. Considerando-se que, atualmente, a Computação e a tecnologia relacionada a esta permeiam todas as demais áreas do conhecimento, a compreensão de programas computacionais é um requisito fundamental na formação dos estudantes em diferentes campos do conhecimento.

A aprendizagem da lógica de programação exige uma base sólida de formação relacionada ao raciocínio lógico e matemático bem como à interpretação e análise de problemas a fim de entendê-los e propor soluções. Neste sentido, defende-se o ensino de conceitos de Computação desde o ciclo da educação básica, pois, segundo os autores Raabe et al. [1], estes são tão importantes para a vida contemporânea

quanto os conhecimentos básicos de Matemática, Física, entre outras ciências.

A motivação para a aprendizagem de conceitos de lógica de programação, desde o ensino básico, tende a ser algo natural para os estudantes da atualidade. De acordo com Prensky [2], esta geração de estudantes cresceu/cresce com fácil acesso à tecnologia de aplicativos gráficos e interativos, pois estão cercados por computadores, vídeo games, smartphones, tablets, entre outros. Como resultado deste ambiente digital onipresente e da interação com o mesmo, os estudantes de hoje pensam e processam informação fundamentalmente diferente de seus antecessores.

Papert [3] complementa esta ideia, destacando que, desde criança, os estudantes aprendem a utilizar a tecnologia de forma hábil e natural, se movimentando com facilidade e rapidez nesse mundo tecnologicado. Esse acesso a uma grande variedade de informações e possibilidades de comunicação e interação passa a fazer parte do seu mundo, da sua cultura e do espaço onde constroem a sua ontogenia. Small e Vorgan [4] afirmam que os cérebros dos estudantes das últimas gerações estão se tornando mais maleáveis e com maior plasticidade, evidenciando uma maior atividade em áreas que afetam a tomada de decisão e o raciocínio complexo.

Assim, o processo de aprendizagem pode se beneficiar da disposição e da familiaridade da atual geração de alunos com ambientes virtuais, jogos digitais e uma extrema demanda pela experiência concreta [5]. Neste contexto, existem diferentes propostas de ferramentas para o ensino de lógica de programação para as crianças como o Scratch [6], Blockly [7], Alice [8], App Inventor [9], Swift Playgrounds [10], entre outras. Tais ferramentas procuram abstrair a codificação de um programa a partir do emprego de recursos visuais, como blocos coloridos que se interconectam e linguagem de programação reduzida, gerando gráficos, animações e/ou jogos simples como resultado.

Nesta perspectiva, aplicações gráficas digitais e interativas podem auxiliar no processo de ensino e de aprendizagem dos conceitos relacionados à lógica de programação. Tal fato se dá, principalmente, devido a tais aplicações estimularem

o auto aprendizado por parte dos estudantes. Aliando-se conceitos de gamificação e/ou jogos digitais a estas, permite-se o desenvolvimento de múltiplas inteligências, uma vez que tais conceitos tornam o aprendizado mais atrativo, divertido e desafiador.

A experiência de uso criada em um jogo, sendo digital ou não, remete a natureza do brincar, representando, desta forma, uma atividade lúdica [11]. Para tanto, considera-se quatro elementos essenciais: representação, interação, conflito e segurança [12] [13] [14]. A representação é o resultado da modelagem para representar o ambiente gráfico onde a ação se desenrola. A interatividade, por sua vez, trata da possibilidade de que ações do usuário retornem reações do sistema, criando um efeito de ação e reação, e permitam a utilização de um jogo por várias vezes para desafiar o usuário e envolvê-lo no contexto do sistema. Já o conflito deve existir no jogo através de desafios e da possibilidade de interagir com o sistema, o que reforça a sensação de imersão em um meio digital. Por fim, a segurança permite que o usuário vivencie psicologicamente o conflito do jogo, experimentando todas as suas opções disponíveis.

Considerando o exposto, este artigo apresenta a proposta de uma ferramenta, intitulada CodeBots, que destina-se ao ensino e à aprendizagem de lógica de programação a partir de um ambiente lúdico e interativo. O objetivo principal é que os estudantes da educação básica aprendam lógica computacional juntamente com conceitos iniciais de Inteligência Artificial de forma lúdica. Para tanto, foi construído um ambiente que retrata uma arena de batalha de bots em um jogo tridimensional (3D). Para que o usuário consiga controlar seu avatar, que também é um bot da arena, este deve inserir instruções, em português estruturado.

A ferramenta proposta foi testada e validada junto a uma turma do sexto ano do ensino fundamental em uma Escola Estadual na cidade de Porto Alegre - RS. Os resultados obtidos indicam que as mesmas se sentiram motivadas a aprender lógica de programação e consideraram divertida e desafiadora a interação com o jogo via codificação, sem utilizar métodos tradicionais como controles, mouses e teclados.

Este artigo está organizado como segue. Além desta seção introdutória, a seção 2 discute trabalhos relacionados com a presente proposta. A seção 3, por sua vez, aborda o projeto CodeBots, apresentando este e suas particularidades. Na sequência, tem-se as seções 4 e 5 que abordam, respectivamente, os testes de validação com a ferramenta proposta e resultados obtidos, com a discussão e principais highlights extraídos destes. Por fim, a seção 6 traz as considerações finais seguida pelos agradecimentos às pessoas que colaboraram com este projeto.

II. TRABALHOS RELACIONADOS

Diferentes autores estão voltando sua atenção para o ensino de lógica de programação desde o ensino básico, mas,

também, no ensino médio e em cursos superiores que não são da área da Computação.

Uma das ferramentas mais utilizadas e precursora na proposta de ensino de lógica, através de blocos gráficos que se encaixam de acordo com instruções lógicas de um código, é o Scratch [6]. O Scratch é um projeto do grupo Lifelong Kindergarten no Media Lab do Instituto de Tecnologia de Massachusetts (MIT) e foi projetado para crianças e adolescentes a partir de oito anos, embora seja usado por pessoas de todas as idades. O Scratch é usado em diferentes países e está disponível em mais de quarenta idiomas, oferecendo versões gratuitas para sistemas operacionais MS Windows, Linux e MacOS). A Figura 1 apresenta uma imagem da interface do Scratch.

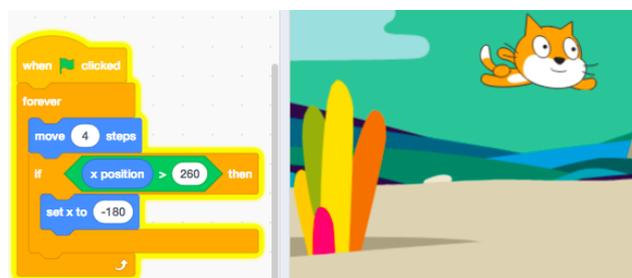


Figura 1. Sequenciamento de operações no Scratch.

Scaico et al. [15] são exemplos de autores que empregam o Scratch em um projeto para o ensino de programação para estudantes do ensino médio. A proposta visa incentivar estudantes a desenvolver habilidades através do pensamento algorítmico com uma abordagem de ensino orientado ao design. O artigo também relata uma olimpíada de programação que serviu para motivar os participantes a empregar os conceitos de lógica de programação aprendidos. Os autores destacam, ainda, que a partir desta experiência, os estudantes despertaram para a possibilidade de seguir a carreira de Computação.

Os autores Oliveira et al. [16] também relatam a experiência de um projeto envolvendo o ensino de conceitos básicos de Ciência da Computação usando o Scratch. No entanto, o projeto focou nos alunos do ensino fundamental/básico e, além do Scratch, foi empregada a metodologia de computação desplugada. Os resultados obtidos mostraram que é possível incluir a temática da lógica de programação no cotidiano escolar a fim de despertar o interesse dos alunos da educação básica.

Em relação ao ensino fundamental/básico, Kologeskiet al. [17] também empregam Scratch, dentre outras ferramentas, na sua pesquisa. Esta descreve o projeto chamado Logicando, desenvolvido por estudantes e professores de uma Universidade em colaboração com escolas locais. O trabalho tem como objetivo desenvolver o raciocínio lógico de alunos dos últimos anos do ensino fundamental,

utilizando ferramentas da Tecnologia da Informação. O projeto utiliza uma abordagem interdisciplinar que relaciona diferentes assuntos com atividades de raciocínio lógico, auxiliando na construção do conhecimento aos alunos da escola básica. Os autores apontam como resultado que, além do desenvolvimento da aprendizagem, o raciocínio lógico proporcionou aos estudantes a experiência do contato com a vida acadêmica, incentivando estes a ingressarem em cursos de Computação no futuro.

Junqueira [18], por sua vez, emprega a ferramenta App Inventor [9] (Figura 2) para ensino de lógica de programação voltada a dispositivos móveis com Android. Os estudantes que participam do projeto são alunos de cursos de graduação nas áreas das Engenharias, os quais apresentam alto índice de reprovação. Os resultados obtidos permitem concluir que o uso de um ambiente de desenvolvimento com ferramenta gráfica voltada para smartphones torna o processo de ensino e de aprendizagem mais próximo da realidade do estudante de graduação, motivando-os na aprendizagem de lógica de programação.

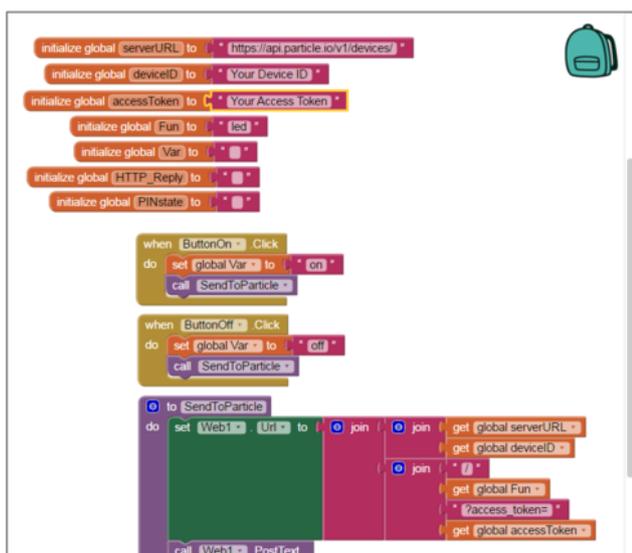


Figura 2. Blocos de comandos no App Inventor.

Bordin e Kepfert [19] também empregam o App Inventor para ensino de lógica de programação junto a estudantes do ensino médio. Os autores analisam o cenário e as percepções dos alunos acerca das oportunidades de acesso à projetos na área da Computação. Os resultados mostram que as oportunidades de acesso a projetos dessa natureza são escassas e que os alunos se sentem mais confiantes em sua capacidade de criar tecnologia ao invés de apenas consumi-la.

Viana e Portela [20] apresentam uma análise do uso de softwares educativos para apoiar a introdução à lógica e algoritmos, tanto no ensino de base quanto no ensino superior. Para tanto, é realizada uma análise de trabalhos

que relatam o uso dos softwares Scratch, Robocode [21] e Visualg [22]. Os resultados desta análise apontaram que alguns desses softwares são mais adequados para o ensino fundamental e médio e outros para a graduação, sugerindo-se propostas de abordagens para adoção dessas ferramentas em sala de aula.

Todas estas abordagens, dentre outras existentes e que não foram citadas nesta seção por limitação de páginas, abordam a questão do desafio do ensino de lógica de programação, considerando ensinamentos básico, médio e superior. No entanto, a ferramenta CodeBots proposta neste estudo se diferencia das demais ferramentas por permitir que os estudantes pratiquem a programação, em português estruturado, ao mesmo tempo em que interagem com um jogo 3D de batalha em arena, aliando-se o lúdico ao aprendizado.

III. O PROJETO CODEBOTS

Atualmente existem diversas tecnologias digitais que tiveram seu desenvolvimento impulsionado pela ascensão do poder computacional e da internet. Estas tecnologias podem ser concretizadas como ferramentas que colaboram com os docentes no desenvolvimento de metodologias e práticas pedagógicas [6] [15] [16] [17] [18]. Através disso, é possível instigar a criatividade, propiciar o desenvolvimento da autonomia, da autoria, da colaboração e da cooperação entre os sujeitos. Marcolino e Barbosa [23] apontam que dentre os diversos fatores que desafiam o ensino da programação, podem ser citados o uso de metodologias tradicionais de ensino que visam a aprendizagem de conceitos dinâmicos utilizando abordagens e materiais essencialmente estáticos. Além deste fator, o uso de linguagens de programação com sintaxes complexas e carência de representações visuais.

A solução proposta neste artigo busca minimizar os problemas e desafios do ensino de programação citados nas seções I e II. No contexto destas tecnologias é possível criar um ambiente de compartilhamento entre professores e estudantes, onde todos devem estar dispostos a ensinar e aprender. Essa perspectiva é apresentada no documento “Diretrizes de implementação” que integra a coleção Padrões de Competência em Tecnologias da Informação e Comunicação (TIC) para Professores, produzido pela UNESCO [24]: Para viver, aprender e trabalhar bem em uma sociedade cada vez mais complexa, rica em informação e baseada em conhecimento, os alunos e professores devem usar a tecnologia de forma efetiva, pois em um ambiente educacional qualificado, a tecnologia pode permitir que os alunos se tornem: usuários qualificados das tecnologias da informação; pessoas que buscam, analisam e avaliam a informação; solucionadores de problemas e tomadores de decisões; usuários criativos e efetivos de ferramentas de produtividade; comunicadores, colaboradores, editores e produtores; cidadãos informados, responsáveis e que oferecem contribuições.

Na conjuntura atual, considerando os desafios de ensino e de aprendizagem perpassados pelo uso de Tecnologias de

Informação e Comunicação, foi proposto o CodeBots, um jogo digital com o objetivo de auxiliar na atividade docente e discente do ensino e aprendizagem básico de programação.

Na solução, conforme ilustra a Figura 3, é possível visualizar diversos robôs e alguns itens coletáveis. No canto esquerdo da tela, a interface apresenta algumas informações sobre todos os robôs: nome, vida e pontuação. Ao lado destas informações, no topo superior direito, estão os botões "Start Battle" para entrar em modo de batalha contra outros robôs, o "Reset" para reiniciar a batalha e o "Quit" para sair do programa.



Figura 3. CodeBots Arena

No lado direito da tela está o elemento que pode ser considerado a parte principal da solução no âmbito da interação: uma caixa de texto onde o estudante pode inserir códigos de programação para definir o comportamento de um dos robôs. Dentre estes comportamentos estão ações como caminhar, virar para esquerda ou para a direita e atirar, além de analisar o ambiente para verificar a melhor ação a ser realizada. Uma característica da solução CodeBots é que, ao receber um comando de programação, as ações ocorrerão de forma instantânea, sem a necessidade de processos intermediários como ocorre em diversos contextos na programação tradicional. Assim, as respostas oriundas das interações dos estudantes ocorrem em tempo real, o que converge com o esperado pelo estudantes, de acordo com as características apresentadas por diversos autores [1] [2] [4] [6].

O CodeBots apresenta dois modos de funcionamento: modo de treino e modo de batalha. No modo de treino é possível modificar o código de programação e definir as ações de um robô. Há outros robôs no cenário, contudo, estes ficam estáticos e não apresentam comportamento ofensivo. Além dos robôs, existem dois itens posicionados aleatoriamente pelo cenário: bombas, que causam danos ao colidirem com o robô, e kits de recuperação, que fazem o robô se recuperar de avarias aumentando seus pontos de vida (que podem ser de 0 até 100). Após desenvolver um código de programação, o estudante pode entrar em modo de batalha, no qual os demais robôs terão comportamento agressivo e tentarão destruí-lo.

No modo de batalha não é possível modificar o código de programação.

Para desenvolver o código, preferencialmente devem ser observadas condições de ações, por exemplo: observar se o robô estiver próximo a uma parede e, a partir desta detecção, definir que ele deve parar de caminhar e desviar do obstáculo. Para isso foi desenvolvida uma linguagem própria para a solução, tendo os principais comandos exibidos na Tabela 1.

Tabela I
BASE DE COMANDOS DO CODEBOTS

Condicional	se
Condições	perto, longe
Objetos	parede, robô, vida, bomba, missil, nada
Ações	caminhar, parar caminhada, virar esquerda, virar direita, parar virada, atirar

Através da combinação desses comandos é possível programar o comportamento do robô de diferentes maneiras, como no exemplo da Figura 4, através do código criado por um discente em um teste preliminar do CodeBots.

```

se longe robo atirar
se perto robo atirar
se perto nada virar esquerda
se longe vida parar virada
se longe vida caminhar
se perto bomba virar direita
    
```

Figura 4. Exemplo de instruções no CodeBots.

A partir da criação de diferentes códigos, é possível que ocorram interações entre os estudantes, inclusive a troca destes códigos para que um batalhe contra o outro. Apesar da tecnologia ter potencial para colaborar na prática docente, ela por si só não será a razão de se obter êxito na atividade, portanto, cabe ao docente problematizar situações para que a mesma tenha um aproveitamento adequado.

O CodeBots possui uma nova versão em desenvolvimento, com gráficos mais atualizados e programação através de ícones, sem a necessidade de digitar os comandos. Nesse novo sistema, a definição das operações do robô segue a mesma lógica da versão apresentada anteriormente, porém, procura facilitar a entrada de dados por parte dos estudantes. A Figura 5 traz a imagem da nova versão gráfica do CodeBots.

A nova versão do jogo possui gráficos mais aprimorados, o que exige um maior poder de processamento dos computadores onde será utilizada. No experimento realizado neste trabalho - que será apresentado na próxima seção - foi utilizada a versão anterior do CodeBots, considerando a aplicação em uma Escola Estadual de Ensino Fundamental e



Figura 5. Versão da programação por ícones.

Médio. A Escola possui um laboratório de informática com computadores oriundos de doações, tendo as máquinas com um poder de processamento relativamente desatualizado para os padrões atuais que os jogos necessitam.

IV. EXPERIMENTO E VALIDAÇÃO

O experimento ocorreu em uma Escola Estadual na cidade de Porto Alegre - RS, em um turma do sexto ano do ensino fundamental.

A. Sujeitos e Procedimento

Participaram do experimento dezesseis estudantes, sendo oito meninos e oito meninas. Todos têm idade entre onze e treze anos e costumam jogar jogos digitais, sendo as principais plataformas PC e mobile.

A atividade de avaliação do jogo proposto ocorreu em três etapas:

- 1) Apresentação do projeto aos estudantes, onde foi explicado o funcionamento do jogo, com detalhamento de mecânicas e objetivos;
- 2) Interação com o jogo por parte dos estudantes;
- 3) Aplicação de questões pós-teste, a fim de avaliar a compreensão dos estudantes em relação aos conceitos básicos de programação.

B. Etapa 1: Apresentação do Projeto

A etapa inicial foi determinada pela contextualização da atividade. Nesta etapa, não foram apresentadas imagens do jogo e não foram utilizados computadores ou qualquer outro recurso digital. Após uma fala sobre contextualização do jogo, foram apresentados os possíveis comandos da linguagem de programação do CodeBots. Essa explanação ocorreu em uma sala de aula tradicional, conforme a Figura 6.

Ainda na primeira etapa, os estudantes foram incentivados a anotar os comandos iniciais e exemplos de ações que poderiam ser utilizadas no jogo. Não foram observados estudantes que não realizaram anotações ou que demonstraram desinteresse na explicação inicial. Enquanto realizavam



Figura 6. Etapa inicial do experimento: Contextualização da atividade.

anotações, diversas perguntas sobre o funcionamento do jogo foram feitas. Após a finalização desta etapa, que ocorreu em aproximadamente quinze minutos, todos os participantes separaram uma folha com exemplos de comandos e se prepararam para sair da sala.

C. Etapa 2: Interação com o Jogo

No início da segunda etapa, os estudantes se deslocaram até o laboratório de informática da escola. A estrutura do laboratório é constituída por onze computadores: um para o professor e os demais para os alunos. Cada partida do jogo foi realizada individualmente, mas alguns participantes tiveram que aguardar um colega finalizar para iniciar o seu turno (Figura 7).

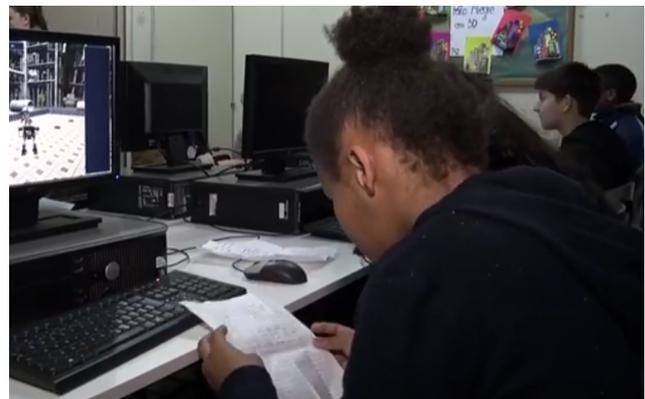


Figura 7. Segunda etapa do experimento: Utilização do CodeBots.

Os estudantes foram instruídos e puderam consultar livremente as suas anotações enquanto jogavam. Ocorreram trocas entre os participantes, principalmente entre os que aguardavam o seu turno de jogo. Observou-se que, mesmo não jogando, os alunos não foram sujeitos passivos, pois incentivavam os colegas e discutiam estratégias para o comportamento dos robôs. Essa etapa ocorreu em, aproximadamente, quarenta e cinco minutos e todos os estudantes

jogaram ao menos duas partidas. Ao terminar a segunda etapa, a atividade foi encerrada.

D. Etapa 3: Questões Pós-Teste

Após a interação dos estudantes com o CodeBots, realizou-se a terceira etapa do procedimento de validação, com a aplicação de um pós-teste constituído por quatro questões. Estas foram propostas de modo a coletar insights sobre a compreensão de noções de programação dos estudantes após a interação destes com o jogo proposto.

Para tanto, um pós-teste, dividido em quatro questões, foi aplicado, sendo três objetivas e uma dissertativa. As três questões objetivas apresentaram cinco alternativas cada uma, sendo que o estudante poderia escolher apenas uma.

A primeira questão pode ser observada na Figura 8. Nesta questão 1, o estudante deveria compreender dois pontos: o primeiro é a possível ação oriunda do comando, que é a rotação do robô à direita.

<p>Questão 1 Analise o comando abaixo, que pode ser usado no CodeBots: <i>se perto parede virar direita</i> Marque somente uma das alternativas: a) Com esse comando o robô sempre vai virar à direita. b) O robô vai virar à direita se existir alguma parede, mesmo que esteja longe. c) Estar perto da parede é a condição para que o robô vire à direita. d) Estar perto da parede é a condição para que o robô ataque. e) Não existem condições no comando, ou seja, o robô sempre vai virar à direita.</p>
--

Figura 8. Questão 1 Pós-Teste

A Figura 9 apresenta a segunda questão do questionário pós-teste. Esta está relacionada ao comando condicional, o qual representa um tipo de instrução fundamental na programação. Espera-se que, mesmo em um curto tempo de interação com o CodeBots, os estudantes compreendam esta questão.

<p>Questão 2 Qual o comando abaixo seria o ideal para desviar de uma bomba? Marque somente uma alternativa. a) se longe parede parar virar b) se perto bomba virar esquerda c) se perto robo desviar bomba d) se longe bomba atirar e) se perto vida virar direita</p>

Figura 9. Questão 2 Pós-Teste

A questão 2 apresenta uma série de alternativas funcionais do CodeBots, questionando qual a mais adequada para que o

robô desvie de uma bomba. Mesmo compreendendo a lógica para escrever os comandos, é preciso que os estudantes avaliem as possibilidades e escolham a opção mais adequada para a estratégia de movimentação do robô. Desta forma, essa questão busca não somente avaliar um comando válido, mas avaliar se a ferramenta proposta está cumprindo o objetivo pretendido.

A terceira e última questão objetiva apresenta três ações que o robô deve realizar: "Atire em robôs que estão perto, pare de caminhar ao ver uma bomba que está longe, vire à direita ao ver uma bomba que está perto". Na sequência, são apresentados comandos para cada uma destas ações, mas com lacunas que o estudante deve completar para escolher a alternativa correta. A Figura 10 apresenta esta questão.

<p>Questão 3 Considere que você quer que o seu robô faça 3 movimentos: - Atire em robôs que estão perto - Pare de caminhar ao ver uma bomba que está longe - Vire à direita ao ver uma bomba que está perto</p> <p>Agora considere os comandos para realizar os movimentos: se _____ robo atirar se longe bomba _____ _____ perto bomba virar direita</p> <p>Marque somente uma das alternativas com as palavras que completam corretamente os comandos acima: a) perto, parar caminhada, se b) longe, caminhar, atirar c) perto, atirar, quero d) longe, parar caminhada, se e) perto, caminhar, se</p>
--

Figura 10. Questão 3 Pós-Teste

Esta questão 3 busca avaliar se o estudante compreendeu que os comandos devem atuar em conjunto para aumentar a complexidade no comportamento do robô.

Por fim, a questão 4 é dissertativa e tem por objetivo perguntar a opinião dos estudantes sobre o jogo (Figura 11).

<p>Questão 4 Escreva sua opinião sobre o CodeBots. O que é legal e o que pode melhorar?</p> <p>_____</p> <p>_____</p>

Figura 11. Questão 4 Pós-Teste

Os participantes levaram entre vinte e trinta minutos para completar a avaliação. Os resultados são analisados na próxima seção.

V. RESULTADOS E DISCUSSÃO

Ao longo da interação dos estudantes com o CodeBots, foi possível evidenciar a motivação e o engajamento destes.

Além da avaliação empírica, realizada a partir da observação dos estudantes enquanto estes interagiam com a ferramenta, foram analisadas as respostas destes no questionário pós-teste aplicado. O gráfico da Figura 12 apresenta os percentuais de acertos e erros, para cada questões pós-teste, por parte dos estudantes.

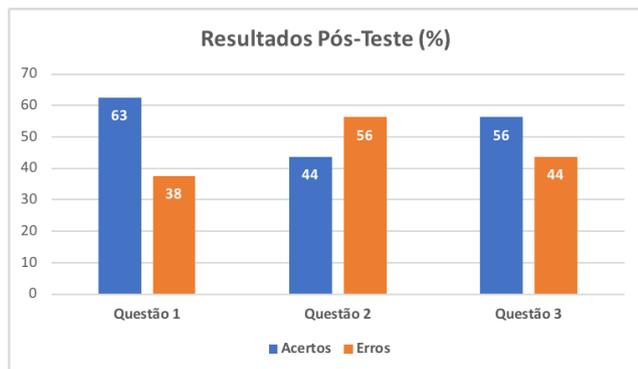


Figura 12. Percentuais de erros e acertos para cada questão pós-teste

A questão 1 foi a que obteve o maior número de respostas corretas: dos 16 participantes, 10 responderam corretamente (aproximadamente 63%) (Figura 13). Apesar do tempo de realização da etapa de interação com a ferramenta CodeBots ser de apenas 45 minutos, conforme exposto na seção anterior, esse resultado aponta que a maioria dos estudantes compreenderam o conceito introdutório de estruturas condicionais em programação. Nota-se, ainda, que os estudantes que responderam de maneira equivocada se concentraram entre as alternativas (a) (12%) e (b) (19%), o que leva a crer que eles começaram a construir o conhecimento relacionado ao comando condicional, já extraindo itens corretos do comando, como objeto (parede) e ação (virar direita).

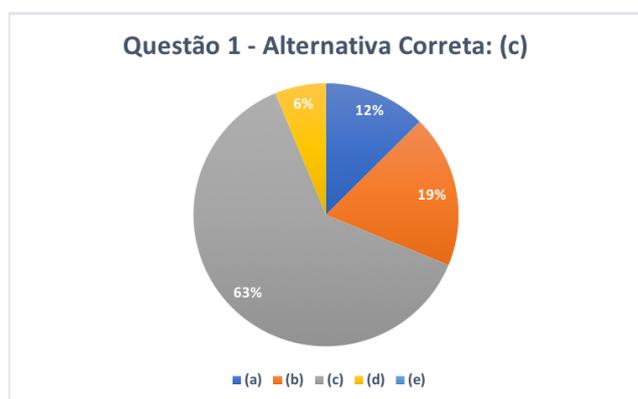


Figura 13. Percentuais de respostas para cada alternativa da questão 1.

Já na questão 2, 7 (44%) estudantes indicaram a alternativa correta (b) enquanto 8 (50%) indicaram erroneamente a alternativa (c) como correta (Figura 14). Essa questão 2

apresentou a particularidade de que, ao contrário da questão 1 em que as respostas incorretas não se concentraram em uma única alternativa, metade dos estudantes escolheu a mesma alternativa incorreta. Tal fato, provavelmente, é decorrente da relação entre o texto da questão e o texto da alternativa (c), onde ambas mencionavam "desviar bomba".

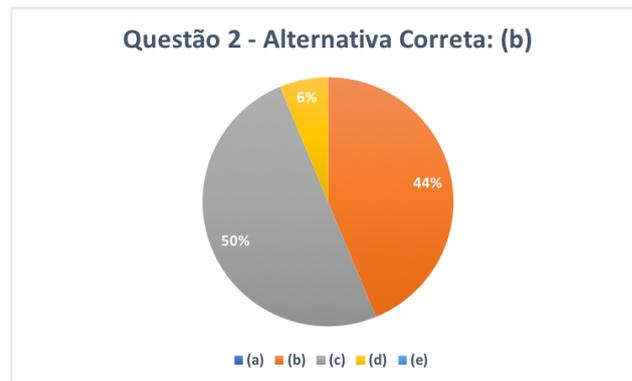


Figura 14. Percentuais de respostas para cada alternativa da questão 2.

Nesta questão, a alternativa correta (b) mencionava "se perto bomba virar esquerda". Apesar disso, a maioria dos estudantes respondeu como correta a alternativa (c), cujo texto era "se perto robô desviar bomba". Nesse caso, o enfoque dos participantes, aparentemente, foi na ação resultante, ou seja, "desviar bomba", desconsiderando a condição que leva a esta. No ponto de vista do aprendizado de lógica de programação, esta é uma questão recorrente quando se trabalha estruturas condicionais: a amarração da condição com os resultados possíveis dentro do contexto do problema a ser resolvido pelo programa.

Porém, mesmo que a resposta (c) não seja a correta em relação à questão 2, considera-se um ponto positivo que os estudantes compreendam que o robô deve desviar da bomba. Considerando que o CodeBots apresenta um número reduzido de instruções, objetos e ações, em uma atividade com mais tempo em relação a este experimento, o equívoco da questão 2 poderia ser melhor trabalhado, fazendo com que alguns estudantes aprendessem com este, que serviria de contra-exemplo em relação à alternativa correta. Esse é um exemplo da necessidade do ensino de programação não focar na construção do "o que", mas, principalmente, do "como" e do "por que".

Já ao analisar os resultados da questão 3, nota-se que a maioria dos estudantes respondeu corretamente, embora 1 estudante não tenha respondido esta questão. Assim, 9 dos 15 respondentes (60%) (Figura 15) marcou a alternativa correta (a). Essa questão teve enfoque em combinar diferentes ações do robô e avaliar o formato correto de escrita e a interpretação de função do comando. Tal como a questão 1, as respostas incorretas não se concentraram em uma única alternativa. Porém, nota-se que, dentre as

incorretas, a alternativa (d) obteve 20% das respostas. Esta era similar à alternativa correta (a), alterando-se apenas a primeira parte da condição: ao invés de "perto", a questão (d) trazia a condição "longe".

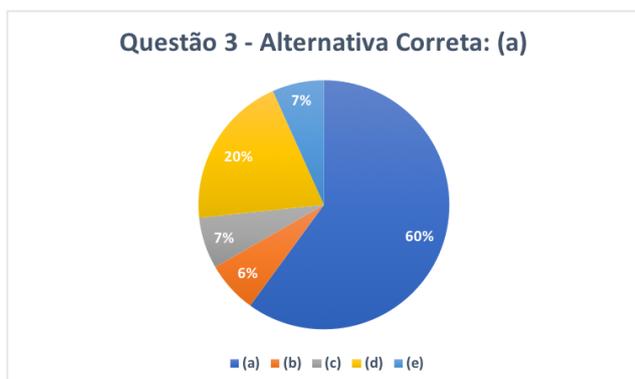


Figura 15. Percentuais de respostas para cada alternativa da questão 3.

De um modo geral, as questões objetivas apontam que a maioria dos estudantes compreenderam os conceitos básicos de programação esperados. A compreensão dos comandos, objetos e ações da ferramenta proposta poderia ser aprofundada com um maior tempo de utilização. Um tempo reduzido de aplicação pode ser considerado válido, reforçado pelas características dos estudantes nativos digitais [2] [4] [5], além da realidade das escolas. A maioria das instituições ainda segue o modelo tradicional de ensino, o que pode ser verificado no local do experimento. Os professores relataram que o laboratório de informática é pouco utilizado e, muitas vezes, os próprios estudantes apresentam resistência em realizar atividades alternativas aos métodos tradicionais, definidos pelos mesmos como copiar o conteúdo do quadro e responder perguntas sobre esse conteúdo.

Considerando essa realidade, iniciativas como a aplicação do CodeBots talvez apresentem maior efetividade com uma curta duração, agindo como um catalisador de mudanças de cultura graduais no ensino e apresentando alternativas aos métodos tradicionais, sem necessariamente buscar uma mudança de forma abrupta, o que poderia inviabilizar tais modificações no modo de ensinar e de aprender.

Por fim, as respostas abertas dadas à questão 4 apontam a motivação dos estudantes ao uso do CodeBots. A maioria escreveu que o jogo é "legal" e que gostou de jogar. No entanto, não existiu um ponto de convergência em relação à percepção geral dos estudantes. Alguns sugeriram que gostariam de jogar no celular, outros de poder personalizar o robô e alguns comentaram que o personagem poderia ser mais veloz. A observação mais recorrente foi o desejo de poder controlar o robô pelo teclado.

Pela proposição do jogo, a principal forma de controle deve ser pela escrita dos códigos. Porém, é possível que, em paralelo ao controle do robô por códigos, exista uma

interação direto e em tempo real, afinal, não permitir a interação do jogador pode ser prejudicial à experiência [14]. No caso da versão do CodeBots usada para o experimento deste trabalho, ao entrar no modo batalha o jogador apenas assiste a partida, ou seja, não pode modificar os códigos e interagir com o robô. Uma possibilidade seria permitir que no modo batalha o jogador interagisse auxiliando o robô previamente programado. Um jogo que adotou uma variabilidade no gameplay para atender a necessidade de interação foi Bomberman 4. Ao contrário das versões anteriores, ao ser derrotado na arena o jogador poderia seguir arremessando bombas e mantendo a interação, assim, mesmo não sendo o vencedor do round poderia continuar jogando, mesmo com limitações.



Figura 16. Modo Arena em Bomberman 4.

A sugestão do controle pelo teclado pode ter sido despertada pela falta de interação decorrente do modo batalha, conforme exposto. Na nova versão do CodeBots (apresentada na seção III), é possível alterar a programação dos robôs mesmo em momentos de batalha, o que tende a atender essa percepção dos participantes.

VI. CONSIDERAÇÕES FINAIS

O ensino e a aprendizagem de programação são desafios nos mais diversos contextos do ambiente escolar, independente da faixa etária e classe social dos estudantes. Conforme apresentado ao longo deste trabalho, diversas iniciativas são aplicadas para que estudantes consigam criar seus primeiros programas e aprofundem o conhecimento neste tópico [6] [15] [16] [17] [18]. Tão grade quanto o desafio do ensino e da aprendizagem da programação, para as mais diversas faixas etárias, é o trabalho realizado com os nativos digitais [2] [4]. Nesse contexto, o uso de jogos digitais pode ter um resultado positivo.

Papert [3] apontava: as crianças podem aprender a usar computadores habilmente e essa aprendizagem pode mudar a maneira como elas conhecem outras coisas. Atualmente o uso de smartphones, tablets e computadores se tornou comum, porém, no caso de muitas escolas do Brasil, não é uma verdade absoluta. Dificilmente algum estudante não terá contato com celulares, por exemplo, mas isso não quer dizer que tenha fluência tecnológica para uso de um computador e que tenha uma base de conhecimento que facilite o início da aprendizagem de programação. Nesse contexto, o uso de programas e jogos desenvolvidos de forma adequada, pensando em experiência do usuário, podem construir uma forma de preencher diversas lacunas de aprendizagem. Assim como jogos, programas de computador e sistemas diversos evoluem ao longo dos anos, o mesmo deve ocorrer com as ferramentas propositivas ao ensino da programação.

Além da falta de infraestrutura e de, em alguns casos, ocorrerem dificuldades na aplicação de metodologias de ensino não tradicionais pela questão cultural dos alunos, despertar nos docentes a necessidade do uso de novas tecnologias e metodologias também é desafiador. Para facilitar esse processo, é importante que o jogo digital consiga ser o mais acessível possível e tenha uma rápida curva de aprendizagem. No caso do experimento realizado neste trabalho, foi preciso uma explanação inicial sobre o funcionamento do CodeBots, conforme exibido na Figura 6. O ideal seria que o próprio jogo apresentasse essas possibilidades aos estudantes. Essa perspectiva está sendo implementada na nova versão. Mesmo com esse aspecto consolidado, a presença do educador é fundamental: ele deve ser um articulador e facilitador, atuando para incentivar os estudantes a trocarem experiências e discutirem diferentes possibilidades e estratégias de programação dentro do jogo.

É preciso que as ferramentas digitais de ensino de programação evoluam juntamente com os padrões tecnológicos vigentes. A vida dos estudantes não se resume ao que eles experimentam e interagem na escola. Atualmente é cada vez mais comum o uso de tecnologias e jogos com elevado realismo gráfico e qualidade através de dispositivos móveis, computadores e consoles. Manter as tecnologias educacionais atualizadas e no padrão dos jogos e programas usados pelos estudantes os torna atrativos. Somente assim serão ferramentas que instigarão os nativos digitais e irão gerar o engajamento necessário ao aprendizado.

De um modo geral, entende-se que colaborar para que os estudantes compreendam a forma de utilizar os comandos em uma linguagem de programação é um dos aspectos que diferentes projetos, como o CodeBots, buscam construir.

AGRADECIMENTOS

Gostaríamos de agradecer aos estudantes e aos professores da Escola Estadual de Ensino Fundamental William Richard Schisler por aplicar o CodeBots e permitir o experimento

no ambiente escolar. Agradecemos ao Centro Universitário Ritter dos Reis pela disponibilidade de recursos para desenvolvimento do projeto e aos discentes participantes: Nathalia Ferraro e Gabriel Isquierdo.

REFERÊNCIAS

- [1] A. L. A. Raabe, A. F. Zorzo, I. Frango, L. Ribeiro, L. Z. Granville, L. Salgado, M. J. K. da Cruz, N. Bigolin, S. A. C. Cavalheiro and S. Fortes, Referenciais de Formação em Computação: Educação Básica, Workshop sobre Educação em Computação, Sociedade Brasileira de Computação (SBC), Jul, 2017.
- [2] M. Prensky, Computer games and learning: Digital game-based learning, in Raessens, J. Goldstein, J.H. (Ed.), Handbook of computer game studies. ed. Cambridge, Mass: MIT Press, 2005, pp. 97-122.
- [3] S. Papert, A máquina das crianças; repensando a escola na era da informática, Porto Alegre: Artes Médicas, 1994.
- [4] G. Small and G. Vorgan, iBrain: Surviving the Technological Alteration of the Modern Mind, New York: Harper Collins, 2008.
- [5] J. Mattar, MATTAR, João. Games em Educação. Como os nativos digitais aprendem, São Paulo: Pearson, 2009.
- [6] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman and Y. Kafai, Scratch: Programming For All, Communications of the ACM, Vol. 52 No. 11, 2009. pp. 60-67.
- [7] E. Pasternak, R. Fenichel and A. N. Marshall, Tips for creating a block language with blockly, 2017 IEEE Blocks and Beyond Workshop (B&B), Raleigh, NC, USA, 2017, pp. 21-24.
- [8] W. P. Dann, S. Cooper and R. Pausch, Learning to Program with Alice (2 ed.), Prentice Hall Press, Upper Saddle River, NJ, USA, 2008.
- [9] D. Wolber, App inventor and real-world motivation, Proceedings of the 42nd ACM technical symposium on Computer science education, Dallas, TX, USA, 2011.
- [10] M. Mathias and J. Gallagher, Swift Programming: The Big Nerd Ranch Guide, 1st ed. United States: Big Nerd Ranch Guides, 2015.
- [11] J. Huizinga, Homo Ludens: o jogo como elemento de cultura, Perspectiva, São Paulo. 2000.
- [12] B. Bates, Game Design, Boston: Course Technology PTR, 2004.
- [13] D. Edery and E. Mollick, Changing the Game: how video games are transforming the future of business, New Jersey: FT Press, 2008, pp. 240.
- [14] J. Schell, The Art of Game Design: A book of lenses, Burlington: Elsevier, 2008.

- [15] P. D. Scaico, E. F. Corlett, L. F. Paiva, E. H. Raposo and Y. Alencar, *Relato da Utilização de uma Metodologia de Trabalho para o Ensino de Ciência da Computação no Ensino Médio*, In: Congresso Brasileiro de Informática na Educação – 18o Workshop de Informática na Escola, Rio de Janeiro. 2012.
- [16] M. L. S. S. Oliveira, A. A. Souza, A. F. Barbosa and E. F. S. Barreiros, *Ensino de lógica de programação no ensino fundamental utilizando Scratch: um relato de experiência*, XXXIV Congresso da Sociedade Brasileira de Computação – CSBC. Brasília/DF. 2014. pp. 1525-1534.
- [17] A. Kolgeski, C. Silva, D. Barbosa, R. Mattos and S. Mioreli, *Desenvolvendo o Raciocínio Lógico e o Pensamento Computacional: Experiências no Contexto do Projeto Logi-cando*, Revista Novas Tecnologias na Educação (RENOTE), v. 14, n. 2, 2016.
- [18] S. H. F. Junqueira, *Ensino de Lógica de Programação com o uso de Ferramenta Visual voltada para Dispositivos Móveis com Android*, XXXVI Encontro Nacional de Engenharia de Produção, João Pessoa/PB, Brasil, 2016.
- [19] A. Bordin and W. E. Kepefrt, *Ensino de programação para o ensino médio com App Inventor: Um relato de experiência extensionista através da colaboração interinstitucional*, In: Anais do Workshop de Informática na Escola. 2018, pp. 205.
- [20] G. A. Viana and C. S. Portela, *O Uso de Softwares Educativos para Introdução de Lógica de Programação no Ensino de Base e Superior*, Informática na Educação: teoria & prática, Porto Alegre, v. 22, n. 1, 2019, pp. 10-22.
- [21] M. Ferreira, R. Filho, F. Rodrigues, S. Sá and J. Silva, *Robocode: um auxílio no aprendizado da linguagem Java*, Em: Encontros Universitários, UFC Quixadá, 2012.
- [22] C. M. Souza, *VisuAlg – Ferramenta de Apoio ao Ensino de Programação*, Revista TECCEN – Vol. 2, no.2; 2009.
- [23] A. Marcolino e E. Barbosa, *Softwares Educacionais para o Ensino de Programação: Um Mapeamento Sistemático*, Maceió: Simpósio Brasileiro de Informática na Educação, 2015.
- [24] Unesco, *Padrões de Competências em TIC para Professores: Diretrizes de Implementação*, 2009.