

Furbot Móvel: um jogo para o ensino do pensamento computacional

Mauro Marcelo Mattos, Luciana P. de Araújo Kohler, Fabrícia D. Zucco,
Andrea Wuo, Bruno F. F. Santos, Joan G. Tridapalli, Heitor Silveira,
Leonardo Fronza, Gian Carlo Giovanella, Liz Rios Largura, Jéssica Maria de Melo
Adrian Kohls, Rafele Caroline Wessling¹

¹ Laboratório de Desenvolvimento e Transferência de Tecnologia (LDTT)
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil

{mattos, lpa, fabricia, awuo, bffsantos, jgtridapalli, hucsilveira}@furb.br

{leofronza, gcgiovanella, lrlargura, jmmelo, akohls, rwessling}@furb.br

Resumo. *O pensamento computacional vem sendo estudado de modo a ser inserido no currículo escolar a partir do ensino fundamental. Dessa forma, este artigo descreve o protótipo Furbot Móvel, sendo este um jogo para o ensino do pensamento computacional disponível para a plataforma Android. A plataforma na sua versão anterior (para desktop) já foi testada com cerca de 190 crianças e este protótipo foi testado em uma turma pequena com 8 crianças. Como resultados, obteve-se que a nova versão do jogo possui maior atratividade e jogabilidade, inserindo o usuário jogador na história do jogo.*

1. Cenário de Uso

O Pensamento Computacional é uma forma de ensinar a resolver problemas baseado em conceitos da Ciência da Computação [Wing 2006]. Contudo, segundo [Nascimento et al. 2018], o pensamento computacional vai mais além, pois ensina como resolver problemas cotidianos baseado em conceitos utilizados há vários anos, como abstração, decomposição, algoritmos e reconhecimento de padrões.

Nesse contexto, a Sociedade Brasileira de Computação (SBC) [SBC 2017], junto com o Currículo de Referência em Tecnologia e Computação (CIEB) [CIEB 2018], vem estudando formas de introduzir o pensamento computacional ainda na matriz curricular da educação básica. Existem alguns estudos que aplicam conceitos de pensamento computacional a partir de dinâmicas em sala de aula utilizando conceitos matemáticos e lógicos e ainda utilizando plataformas de programação disponibilizadas como o Scratch e o Code.org. Acontece que, o pensamento computacional pode ser aplicado em diversas áreas do saber, além das áreas exatas, pois em todas as áreas existem problemas a serem resolvidos e acerca disso a plataforma Furbot induz o jogador à trabalhar sua capacidade de decomposição e abstração de problemas. Em relação as plataformas existentes, de forma geral, não trabalham com conceitos de jogabilidade, sendo ferramentas com aspectos lúdicos, mas com o objetivo principal de ensinar a programação. Ainda, essas ferramentas normalmente trabalham com programação em blocos de modo que o usuário que a utilize seja levado a uma construção abstraída de funções e rotinas, obstruindo a compreensão total do funcionamento do código.

Vendo essa problemática, desenvolveu-se a plataforma Furbot que tem por objetivo introduzir o pensamento computacional a partir de um jogo que possui um enredo

próprio para introduzir conceitos computacionais de forma lúdica. Essa plataforma resultou em alguns produtos, sendo um deles o jogo Furbot Móvel, o qual será apresentado neste artigo. O jogo pode ser trabalhado com crianças a partir do primeiro ano do ensino fundamental até o público adulto. A medida em que a criança aprende novos comandos e desenvolve seu raciocínio lógico, podem ser introduzidos novos conceitos e recursos no jogo, de modo que uma mesma fase pode ser resolvida de diferentes formas. Isso porque o jogo disponibiliza de várias funções que combinadas resolvem os exercícios, sendo possível utiliza-las de diferentes formas. Um exemplo disto é a introdução de condicionais para fazer verificações sobre o cenário (como obstáculos). Ainda, os laços de repetição que proporcionam uma percepção além de somente resolver o problema, mas também de fazê-lo da forma mais eficiente possível por permitir a redução na quantidade de instruções à serem escritas para a resolução de problemas.

2. Desenvolvimento

Originalmente, o Furbot é um *framework* desenvolvido em Java que é utilizado para ensinar programação para os alunos dos cursos de Ciência da Computação e Sistemas de Informação da Universidade Regional de Blumenau (FURB) [Vahldick and Mattos 2008]. Em 2016, a partir do *framework*, decidiu-se criar uma plataforma lúdica para o ensino do pensamento computacional e aplica-la com crianças de ensino fundamental (anos iniciais) [Mattos et al. 2018].

Para o desenvolvimento dessa plataforma lúdica, uniu-se um grupo de professores e alunos de graduação dos cursos de Ciência da Computação, Sistemas de Informação, Publicidade e Propaganda, Pedagogia e Matemática formando assim uma equipe interdisciplinar. Nessa equipe, cada área tratou de um tema específico relacionado a sua competência, como: programação; leiaute da plataforma; enunciado das atividades; aplicação em sala de aula; entre outros. Após desenvolver várias práticas em sala de aula em duas escolas estaduais da cidade de Blumenau ao longo de dois anos [Araújo et al. 2018], decidiu-se redesenhar a plataforma, transformando-a em um jogo para o ensino do pensamento computacional. Optou-se pela modificação da plataforma para que a mesma adote o formato de um jogo, mas que se mantivesse os conceitos fundamentais da plataforma acerca do pensamento computacional. Dessa forma, além do jogo ter um enredo que familiariza os jogadores com o contexto do jogo e dá um propósito para suas ações, a progressão das habilidades e recursos disponíveis no jogo também tornam a experiência mais lúdica e recompensadora de modo que o esforço que o jogador faz em cada fase traduz em uma melhor pontuação.

Dessa forma, o novo Furbot foi desenvolvido na plataforma Unity utilizando a linguagem C#. Para a construção e definição de sua jogabilidade utilizou-se de técnicas de *game design*, sendo os documentos *game treatment* e *high concept*. O *high concept* é um documento que define de forma breve conceitos do jogo como a motivação, gênero, competitividade, pontuação, requisitos de hardware e objetivos [Adams 2008]. Já o *game treatment* define de forma mais detalhada essas informações, além de toda a jogabilidade, como enredo, personagens e outros fatores importantes de um jogo [Claypool and Moriarty 2010].

Esses documentos, bem como todo o desenvolvimento do jogo, foram realizados utilizando técnicas de metodologias ágeis, como o Kanban e o Scrum. Para a definição

dos documentos, a equipe de desenvolvimento realizou reuniões diárias durante duas semanas. Já para a implementação do jogo, a equipe realiza reuniões semanais. Nessas reuniões define-se uma *sprint* com objetivos bem definidos e então dividem-se as tarefas para cada membro da equipe de desenvolvimento e *design*. Na reunião seguinte, cada membro apresenta o que realizou, as dificuldades que teve e se discute a nova *sprint* que será apresentada até a próxima semana. Com essa prática, pode-se obter versões estáveis em pequenos intervalos de tempo.

Como o jogo ainda está em fase de desenvolvimento, tem-se uma versão composta por cinco fases, sendo estas as fases introdutórias do pensamento computacional. Para testar o jogo até o momento, o mesmo foi aplicado com 8 crianças entre 9 e 11 anos que já utilizavam a plataforma Furbot Java [Mattos et al. 2018]. Essas crianças são alunas da turma de programação com o Furbot da Associação Criança em Primeiro Lugar (ACPL), a qual disponibiliza aulas de programação uma vez por semana por um período de uma hora, sendo que o curso já acontecia há 2 meses. Para o teste, as crianças receberam o jogo Furbot Móvel em um dispositivo *smartphone* Android e foram instruídas a jogá-lo. No momento do teste, o jogo ainda não apresentava a seção de tutorial. Logo, ensinou-se como utilizar o ambiente para que então cada uma pudesse passar as fases do mesmo. Após passarem as fases, as crianças foram questionadas em relação a jogabilidade e ainda se preferiam o Furbot Java ou o Furbot Móvel, este desenvolvido em Unity. Todas as oito crianças mencionaram que preferem o Furbot Móvel, pois seu leiaute é mais atrativo, semelhante aos jogos que estão acostumados a jogar. Elas ainda mencionaram que este parece um jogo e dá mais vontade de utilizá-lo do que a versão em Java, que não apresentava essas questões de jogabilidade de forma tão específica.

3. Apresentação do software

O Furbot Móvel é um jogo de aventura e ficção em 2D, linear e sequencial, baseado em níveis. Além do tema aventureiro, o intuito disto é o ensino do pensamento computacional, lógica de programação e conhecimentos gerais de maneira imperceptível. Atualmente uma versão beta está disponível para dispositivos Android ¹ na licença Creative Commons Atribuição-SemDerivações-SemDerivados 3.0 Brasil (CC BY-NC-ND 3.0 BR).

O enredo do jogo se passa no Planeta Terra, o qual foi invadido por alienígenas chamados Buggiens que gostam de poluir planetas. A missão do personagem principal, sendo este o Furbot, é salvar o planeta da invasão Buggien, mantendo-o limpo e preservando o meio ambiente. O enredo pode ser visualizado no início do vídeo demonstrativo² ou no link ³. No contexto do jogo, o Furbot é uma invenção da professora Sam, sendo esta outra personagem do jogo, e é programado por meio de comandos pré-definidos para cumprir missões e encontrar objetos espalhados nas fases. A professora Sam é ilustrada na Figura 1 (A).

A professora Sam foi desenvolvida pela equipe de *design* do projeto. Decidiu-se que seria uma professora, para demonstrar que as mulheres também podem ser inseridas no contexto da programação, uma vez que o jogo ensina a programação por meio do pensamento computacional. Ainda, optou-se por ser uma professora negra, trazendo a

¹<http://www.furb.br/ldtt/arquivos/furbot.apk>

²<http://www.furb.br/ldtt/arquivos/VideoDemo.mp4>

³<http://www.furb.br/ldtt/arquivos/Video.mp4>



Figura 1. (A) Professora Sam (B) Tela inicial

diversidade e a inclusão social para dentro do jogo, de modo que as crianças possam se sentir representadas por esta figura.

Ao abrir o jogo, será exibida a tela inicial conforme pode ser visualizada na Figura 1 (B). Nesta tela, tem-se alguns personagens do jogo, sendo: o Buggien, o alienígena na cor verde localizado a esquerda da tela, que invadiu a Terra para poluí-la; o Furbot, localizado no lado direito da tela, sendo este o personagem principal do jogo que é programado pelo jogador; a S-223, localizada no lado superior direito, sendo um *drone* que guia o Furbot durante as fases, apresentando dicas e informações importantes. Ainda, a figura apresenta um tesouro, sendo este um elemento coletável do jogo. O tesouro dá pontos que podem ser trocados por vidas ou energia.

Após clicar no botão `JOGAR`, o jogador é encaminhado para a cena de tutorial, conforme pode ser visualizado na Figura 2. Nessa cena são apresentados todos os elementos da interface gráfica para que o jogador saiba como manipulá-la durante as fases. Para apresentar a interface, a S-223 aparece indicando na tela e apresentando o significado de cada uma das partes. O tutorial foi desenvolvido para simular uma projeção, na qual o jogador entra no laboratório da professora Sam e visualiza sua missão do jogo. Para iniciar as fases é obrigatório que o jogador tenha assistido o tutorial, sendo que ele não pode pular esta etapa. Caso seja a segunda vez que entrou no jogo, ele poderá ir direto para a seleção de fases. Contudo, como ainda é um protótipo, esta funcionalidade não foi implementada.



Figura 2. Tutorial do cenário

Na tela de tutorial, conforme visualizado na Figura 2 é apresentada uma cena de exemplo. Essa cena contém um caminho que deverá ser percorrido pelo Furbot. Para

completar as fases, o Furbot terá que coletar pistas no caminho que podem ser tesouros ou sujeiras deixadas pelos Buggiens. Ainda, conforme a fase, podem existir outros elementos a serem coletados, como vidas e baterias para recarregar a energia do Furbot. A medida em que o Furbot anda pela cena, ele perde energia. Se andar sobre o caminho, perderá menos energia do que se andar fora dele. Dessa forma, o jogador deve pensar antes de realizar os movimentos do personagem para que seja possível completar a missão com a energia fornecida. A energia atual pode ser visualizada na barra inferior representada pelo segundo ícone com a cor verde. Ainda, quando o Furbot perder toda a energia ou bater em obstáculos como árvores ou lagoas, ele perde uma vida. Ao todo são liberadas três vidas para o jogador que podem ser recarregadas a medida em que o mesmo coleta novas vidas pelas fases. A quantidade de vidas atuais pode ser visualizada na barra inferior representada pelas cabeças do personagem Furbot. Ainda, na barra inferior tem-se o desenho de um tesouro que indica quantos tesouros foram coletados na fase até o momento. Por fim, tem-se uma estrela que indica a pontuação. A pontuação é baseada na quantidade e nos tipos de coletáveis encontrados. Na Figura 2 também pode-se visualizar a área de programação do Furbot, localizada no lado direito da tela. Nela, serão incluídos os códigos para o personagem andar pela cena.

Após finalizar o tutorial, o jogador é encaminhado para a tela de seleção de fases, conforme ilustra a Figura 3 (A). Até o momento tem-se somente uma tela de seleção. Contudo, o jogo terá um total de 4 telas de seleção contendo 5 fases em cada. Em cada tela de seleção será demonstrada uma área diferenciada, sendo que a primeira fase representa a Floresta Amazônica, a segunda representará o Egito, a terceira o Polo Norte e a quarta será dentro da nave Buggien. Assim, o jogo introduz conceitos de História e Ciências que são trabalhados dentro de cada uma das fases. Como por exemplo, na terceira fase deste mapa são apresentadas as vitórias régias e é informado para o jogador que a vitória régia é a maior planta aquática do planeta. Desta forma, além de conceitos do pensamento computacional, o jogador recebe conhecimentos gerais que são ensinados de forma imperceptível.



Figura 3. (A) Seleção de fases (B) Comandos

Para selecionar uma fase, o jogador deve utilizar as setas localizadas no canto inferior direito, sendo que não poderá selecionar a fase 2 antes de ter concluído a fase 1 e assim sucessivamente. A fase de Laboratório retorna para o tutorial e para ir à fase 1, deve-se clicar na segunda seta (com a cor cinza) e então clicar na seta verde. Feito isso, o jogador será encaminhado para a fase 1 como representa a Figura 3 (B).

Todas as fases são introduzidas pela S-223, sendo esta a ajudante do Furbot que indica o que o jogador terá que fazer. Para fechar o balão da S-223 e passar para o próximo, pode-se clicar sobre o mesmo. Nessa primeira cena, o jogador deve sair de sua posição e andar pelo caminho até encontrar a pista na cor verde deixada pelos Buggiens. Esta pista se encontra sobre o caminho mais ao lado direito da tela. Nessa cena, deve-se cuidar para não colidir com as árvores, os arbustos e também para não sair da trilha do caminho, pois não haverá energia suficiente caso o jogador decida andar pelo gramado.

Para fazer com que o robô se movimente são utilizados comandos de programação pré definidos. Ao clicar sobre a área de programação (parte branca à direita) é apresentada a tela de comandos, conforme ilustrado também na Figura 3 (B). Como a digitação em dispositivos móveis é um pouco limitada, decidiu-se construir uma interface com botões que disponibilizam os comandos que fazem o Furbot andar. Os botões devem ser combinados entre si para formar os comandos possíveis do jogo. Por exemplo, para fazer o Furbot andar para baixo, deve-se digitar o comando `andar (abaixo) ;`. Para andar em outra direção, deve-se trocar `abaixo` pela direção desejada. Dessa forma, combinando vários comandos de andar para uma direção, o Furbot será movimentado pelo cenário. Para iniciar o movimento, deve-se clicar no botão `Executar` localizado abaixo da tela de comandos. Caso queira executar todos os comandos do início do bloco, deixa-se marcado o `checkbox` `Recomeçar`. Caso queira andar somente a partir dos novos comandos adicionados, pode-se desmarcar esta opção.

Além do movimento de andar para uma direção, o jogo disponibiliza do comando enquanto. Contudo, o comando enquanto será introduzido para o jogador somente a partir da segunda cena de seleção de fases. No entanto, ele está disponível nessa versão para verificação de sua existência. Para utilizar o comando enquanto deve-se combinar os botões até que se tenha o seguinte código: `enquanto (eh (caminho, direita)) {`. Nesse exemplo, o Furbot executará os passos que estarão entre o abre chaves e o fecha chaves enquanto tiver caminho na sua direita.

Para apagar comandos há duas possibilidades: apagar todos os comandos clicando no primeiro botão `Apagar todos comandos`; ou clicar no segundo botão `Apagar último comando adicionado` que apaga comando por comando de trás para frente. No momento, não é possível apagar uma linha específica, sendo esta uma limitação temporária da versão.

Ainda, quando é executado o código-fonte para que o Furbot ande no cenário, caso houverem erros na combinação dos códigos, a S-223 aparece e informa o problema. Já quando o código-fonte está correto, a linha que está sendo executada pelo Furbot no momento fica em negrito, demonstrando assim o passo a passo realizado. Dessa forma, introduz-se conceitos de um Ambiente de Desenvolvimento Integrado (IDE) por meio de conceitos de um compilador de código-fonte.

Ao finalizar a fase, é apresentada a tela de sucesso com a pontuação adquirida e a quantidade de tesouros coletados. Nessa tela, tem-se a possibilidade de voltar a tela de seleção de fases, reiniciar a fase ou ir para a próxima fase.

Optando-se a ir para a próxima fase, a mesma será carregada diretamente. A Figura 4 demonstra a segunda fase do jogo. Nessa fase, como pode-se observar, existem alguns coletáveis sobre o caminho. O primeiro coletável é a sílaba `EN`, em seguida tem-se

um tesouro e pode-se coletar antes da pista final uma bateria e ainda outro tesouro. Ao coletar a sílaba EN a S-223 irá apresentar seu significado. De forma geral, a sílaba faz parte da palavra “Enquanto”. Desta forma, nas fases da cena de seleção 1 serão disponibilizadas as sílabas para compor este novo comando. Quando o jogador coletar a última sílaba, sendo esta “to”, o comando enquanto será liberado para o jogador. Lembra-se que neste protótipo o comando já vem liberado desde a fase inicial. Os tesouros no caminho são utilizados para o jogador receber pontos. A partir da junção de pontos, o mesmo poderá trocar por vidas e energias na loja do Furbot que é acessada por meio da tela de seleção de fases. Já a bateria serve para recarregar a energia. A ideia é que tanto as vidas, quanto a energia sejam mantidas de uma fase para outra, ou seja, se o jogador terminar com a energia em 50% na fase 1, na fase 2 ele iniciará com 50% também. Dessa forma, o jogador terá que gerenciar sua energia durante todas as etapas do jogo, sendo ensinado a planejar para então tomar a ação.



Figura 4. Segunda fase

Como mencionado anteriormente, essa versão beta possui 5 fases que são situadas na Floresta Amazônica. Quando todas as fases forem liberadas, o final do jogo será com o Furbot resgatando o Planeta Terra da invasão dos Buggiens.

4. Considerações finais

Este artigo demonstrou o protótipo Furbot Móvel desenvolvido em Unity para plataforma Android. A partir de alguns testes realizados com crianças que já utilizaram a plataforma Furbot anteriormente, percebeu-se que esta versão as atrai mais devido a suas questões de jogabilidade e sua interface gráfica atraente.

O jogo traz como contribuições pedagógicas a possibilidade de inserir o pensamento computacional em disciplinas de ensino fundamental, integrando com a própria disciplina. Isso porque o jogo é intuitivo e possui dicas desde o início até o final do mesmo, ensinando o usuário sobre como utilizá-lo. Ainda, o professor pode utilizar o cenário do jogo para trabalhar conceitos de sua disciplina junto com o pensamento computacional. Por exemplo, pode trabalhar conceitos de Ciência no mapa da Floresta Amazônica. Já na disciplina de Matemática o professor pode trabalhar conceitos de vetores e matrizes, ou mesmo de tabela. Ainda, pode trabalhar com capacidades motoras como as direções e a escrita das palavras.

De forma geral, o jogo apresentado se enquadra em várias diretrizes do currículo recomendado pelo CIEB [CIEB 2018] para a inclusão do pensamento computacional na

educação básica, ensinando conceitos de abstração, algoritmos, decomposição de problemas e reconhecimento de padrões por meio das várias fases. Dessa forma, o jogo mostra-se relevante para a área da educação, principalmente voltado a inclusão do pensamento computacional na educação básica.

Esta versão do jogo ainda não foi testada em escolas. Contudo, a versão anterior do Furbot (plataforma em Java), foi utilizada em duas escolas da cidade de Blumenau em aproximadamente 85 oficinas das quais passaram cerca de 190 crianças. A plataforma foi utilizada durante uma hora das aulas do ensino básico, sendo aplicadas em turmas de 1º ao 5º ano. Após as atividades realizadas, percebeu-se uma evolução das crianças em relação a algumas habilidades motoras, bem como em relação ao raciocínio lógico e a forma de resolver problemas. Os resultados são relatados em [Mattos et al. 2018] e [Araújo et al. 2018]. Logo, acredita-se que com uma versão móvel com características mais próximas a um jogo, as crianças queiram utilizar a aplicação também em suas casas, aumentando assim seu aprendizado em relação ao pensamento computacional.

Como projetos futuros pretende-se finalizar o jogo com todas as fases conforme relatado neste artigo. Além disso, pretende-se incluir uma funcionalidade para o jogador criar seus próprios cenários e jogá-los. Dessa forma, professores também poderão criar fases específicas para seu conteúdo em sala de aula. Pretende-se também validar esta versão em turmas de ensino básico, bem como disponibilizar o aplicativo em lojas para *download* de forma gratuita.

Referências

- Adams, E. (2008). The high concept document. In *Välkommen till KTH*.
- Araújo, L. et al. (2018). Ensino do pensamento computacional em escola pública por meio de uma plataforma lúdica. In *Anais dos Workshops do VII Congresso Brasileiro de Informática na Educação (CBIE 2018)*. SBC.
- CIEB (2018). Currículo de referência em tecnologia e computação. Disponível em: <http://curriculo.cieb.net.br/curriculo>.
- Claypool, M. and Moriarty, B. (2010). The game development process. Disponível em: <https://web.cs.wpi.edu/imgd1001/a10>.
- Mattos, M. et al. (2018). Uma pesquisa-ação sobre o desenvolvimento do pensamento computacional com crianças. In *Anais do XXIV Workshop de Informática na Escola (WIE 2018)*. SBC.
- Nascimento, C., Santos, D. A., and Tanzi, A. (2018). Pensamento computacional e interdisciplinaridade na educação básica: um mapeamento sistemático. In *Anais dos Workshops do VII Congresso Brasileiro de Informática na Educação*. SBC.
- SBC (2017). Referenciais de formação em computação: Educação básica. Disponível em: <http://www.sbc.org.br/noticias/10-slideshow-noticias/1996-referenciais-de-formacao-em-computacao-educacao-basica>.
- Vahldick, A. and Mattos, M. (2008). Relato de uma experiência no ensino de algoritmos e programação utilizando um framework lúdico.
- Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3):33–35.