

## Quindim auxiliando na aprendizagem do pensamento computacional

Cynthia M. N. Ungefehr<sup>1</sup>, Raimundo C. S. Vasconcelos<sup>1</sup>

<sup>1</sup>Instituto Federal de Brasília (IFB) – *campus* Taguatinga  
QNM 40, Área Especial nº 01, Taguatinga/DF - 72146-050 - Brasília - DF - Brasil

`cynthia-mn@hotmail.com, raimundo.vasconcelos@ifb.edu.br`

**Resumo.** *Este artigo apresenta um protótipo de um jogo livre denominado Quindim para ensino do Pensamento Computacional (PC), com o objetivo de contribuir para a formação de estudantes do primeiro ano do ensino médio. O jogo foi desenvolvido no motor gráfico Unity Engine com apoio da linguagem de programação C#. Atualmente o jogo tem versões para Android e Windows.*

### 1. Justificativa pedagógica para o desenvolvimento do protótipo

O Pensamento Computacional (PC) é considerado como uma distinta forma de pensar e perceber situações para resolver problemas, essencial para todos no século XXI [Wing 2006]. Ele é útil para uma ampla gama de pessoas, pois seu foco está na resolução eficiente de problemas, algo com que todos têm que lidar diariamente, inclusive alunos do ensino médio. Por conta disso, esse estudo investiga a viabilidade de ensinar pensamento computacional para adolescentes e pré-adolescentes.

Considera-se o PC como imprescindível para a formação de um cidadão e já está presente como componente curricular obrigatório no ensino básico em diversos países como Inglaterra, Estados Unidos, Israel, entre outros [Valente et al. 2017]. No Brasil está inserido como eixo transversal dos componentes curriculares por meio da nova Base Nacional Comum Curricular [MEC 2017], destacado na competência 5; dentre várias habilidades que os alunos precisam desenvolver, consta a linguagem da programação elencada com a compreensão, escrita e representação de algoritmos na resolução de problemas. Esta habilidade também está presente no Currículo de Referência em Tecnologia e Computação [CIEB 2018], juntamente com a cultura e a tecnologia digital.

Nesse sentido é que se propõe um objeto de aprendizagem: protótipo de um jogo para o ensino do pensamento computacional, em formato de aplicativo livre, para dispositivos móveis com sistema operacional *Android*. Com um propósito em mente, diversos trabalhos com objetivos ou perspectivas semelhantes foram analisados. Deles, alguns se destacaram:

O artigo apresentado por [Turchi et al. 2019] consiste na criação de um jogo - “TAPASPlay” - desenvolvido a partir do TAPAS (*T*Angible *P*rogrammable *A*ugmented *S*urface ou Superfície Aumentada Programável Tangível) com uma abordagem construtivista. A partir de quebra-cabeças e restrições, os jogadores podem se reunir em equipes para construir espadas e batalhar contra outras equipes, exercitando o pensamento algorítmico e a abstração em um formato que permite tanto a colaboração, quanto a competição.

O TAPASPlay foi testado por 18 alunas inglesas de 15 anos de idade que possuíam pouca ou nenhuma experiência prévia com programação. Os resultados obtidos

foram considerados positivos: o jogo teve uma recepção positiva das usuárias e conseguiu promover as habilidades de Pensamento Computacional propostas.

Já o trabalho apresentado por [Forquesato 2018], aborda o desenvolvimento do jogo *Crocro's Adventure*, que consiste em um jogo 3D de criação de algoritmos com blocos voltado para o público infantil, no qual a criança ajuda Crocro a encontrar seus doces perdidos em diversas ilhas num total de 41 níveis. Utilizando o *stealth assessment*, o jogo coleta dados sobre o desempenho e ações do jogador, como: quantidade de tentativas e tempo necessário para passar o nível.

O jogo foi testado com um grupo de 29 crianças com idades entre 7 e 11 anos. Após análise dos resultados obtidos, o autor concluiu que o experimento teve sucesso: as crianças gostaram do jogo, provando que jogos educativos podem ser divertidos; também foi verificado que crianças abaixo de 8 anos de idade têm mais dificuldade para entender a abstração e modularização necessárias para resolver níveis mais difíceis.

Quanto ao *stealth assessment*, concluiu-se que é possível conseguir uma boa base de dados que permite verificar não só a evolução do jogador como identificar as crianças que estavam com dificuldades.

Por último, tem-se o trabalho de [Kazimoglu et al. 2012], no qual foi proposto o desenvolvimento de um jogo, “*Program Your Robot*”, de quebra-cabeça no qual o jogador deve guiar um robô até o ponto de teleporte por meio de instruções finitas.

O jogo foi testado com diversos alunos do ensino médio e calouros da faculdade, reunindo 25 feedbacks dos quais foi tirada a conclusão de que o jogo é adequado para ajudar alunos a entender construções introdutórias de programação além de compreender os conceitos de Pensamento Computacional propostos.

Com os resultados das pesquisas em mente, foi escolhida a abordagem construtivista [Papert 1986] para gerar a evolução do aprendizado e a abordagem furtiva (*stealth assessment*) para a avaliação da aprendizagem.

Por questões de mobilidade e visando a implementação também em ambiente escolar, o jogo foi inicialmente pensado para um ambiente móvel, contudo, em vista do período atual de pandemia e resultado de pesquisas de mercado do *Unity Dashboard*, foi decidido também ampliar o alcance do jogo por meio da criação de uma versão para *Desktop*, para dispositivos com sistema operacional *Windows*.

A avaliação e a coleta de dados serão realizadas por meio da avaliação furtiva e, dessa maneira, enquanto os participantes jogam, o aplicativo coletará estatísticas de uso, como: tempo de resposta, quantidade de erros e tentativas e o código criado pelo jogador para análise do progresso e entendimento, que serão utilizadas para determinar se houve aprendizado.

Com isso, tanto a ideia de usar avaliação furtiva para verificar o aprendizado de pensamento computacional, como o jogo como uma ferramenta viável para atingir esse objetivo, serão validados.

## **2. Disciplinas ou Contexto para as quais o protótipo se destina**

O protótipo do jogo se destina a desenvolver as habilidades relacionadas ao pensamento computacional e, nesse sentido, pode-se citar: desenvolvimento do raciocínio lógico,

resolução de problemas, representações e construções de sequências estão relacionadas.

Áreas relacionadas à Matemática, Geometria, Linguagens e Interpretação de problemas são disciplinas/áreas relacionadas à participação no jogo.

### 3. Público alvo do protótipo

Foi escolhido como público inicial para uso do protótipo alunos do primeiro ano do Ensino Médio Técnico (Eletrotécnica e Moveleira). Os alunos tiveram oportunidade em participar do jogo, responderam questionários pré e pós jogo, para uma melhor avaliação.

Mesmo assim, considera-se que este jogo possui um público-geral amplo, podendo ser utilizado por estudantes das séries finais do Ensino Fundamental, ser aplicado em sala de aula, durante aulas de ciências ou matérias relacionadas a matemática e também como jogo livre, fora do contexto formal de ensino.

### 4. Processo de desenvolvimento adotado

Para o desenvolvimento do protótipo foi inicialmente feito um *brainstorm* de ideias com três pontos chave em mente: pensamento computacional, diversão e originalidade. Por fim, decidiu-se por um jogo de plataformas em que, para se avançar na história, seria necessário resolver desafios dados por personagens que fazem parte do jogo e ou cenário, não controlados pelo jogador, conhecidos como NPCs (*Non-Player Character*). Os desafios possuem formato semelhante a diagramas de blocos e permitem que o jogador conecte blocos com funcionalidades diferentes e, com isso, crie algoritmos para resolver o problema dado.

O personagem principal é um gato amarelo com nome Quindim e que também é o nome do jogo. A Figura 1 apresentam a evolução do personagem.

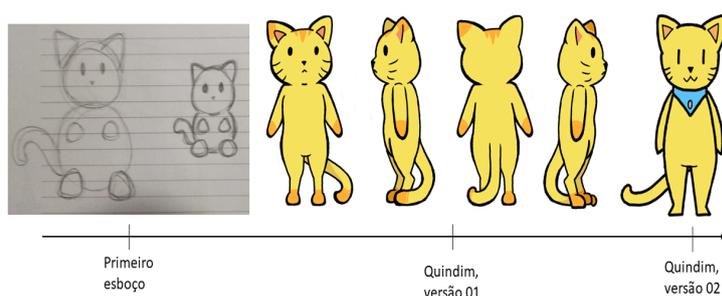
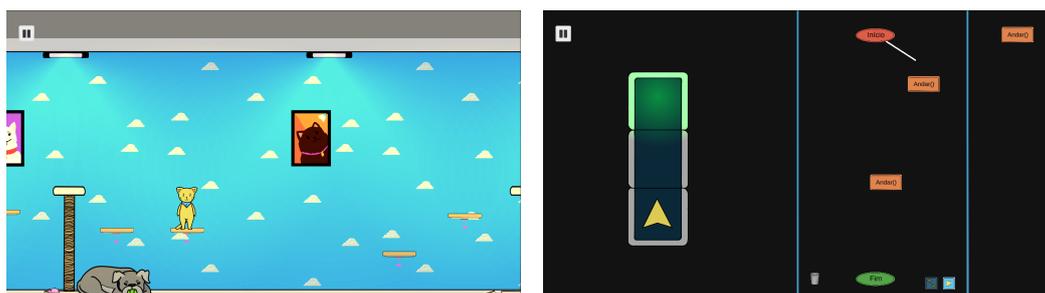


Figura 1. Evolução do personagem Quindim

Com esses objetivos definidos, foram feitas pesquisas para determinar os materiais que seriam utilizados para o desenvolvimento. Decidiu-se por utilizar o popular motor de jogos Unity [Technologies 2021], devido a grande quantidade de material didático gratuito, tanto no *site* oficial do motor, bem como vídeos e tutoriais disponibilizados por criadores de conteúdo em plataformas de *stream* (YouTube) [Brackeys 2020] [CodeMonkey 2021] [CocoCode 2021] [GameLoot 2021] e a facilidade que traria para a configuração do executável para dispositivos móveis e *desktop*.



(a) Nível 01: PetCare

(b) Atividade do tipo "caminho" proposta por Manchas, o cachorro

**Figura 2. Primeiro nível e primeiro desafio do jogo**

## 5. Linguagem de programação utilizada

Foi utilizada para a criação do jogo a versão estável 2020.3.1f1 junto com o Visual Studio 2019 [Studio 2021]. O motor de jogos Unity, desde a versão 2017.1, aceita apenas a linguagem de programação C#, logo essa foi escolhida para o projeto.

Para a criação das artes do jogo foi utilizado o programa de edição gráfica Krita [Foundation 2021] e o pacote *Kenney Game Assets* [Kenney 2020] disponível para a compra na loja de jogos virtual *itch.io*. Os áudios do protótipo foram retirados tanto do pacote mencionado acima quanto de *sites* que disponibilizam os efeitos gratuitamente e editados no programa de edição de áudio *Audacity* [Team 2021].

Tendo em vista a necessidade de avaliar a participação dos alunos, para colocar em prática a avaliação furtiva, foi feita a integração do protótipo com *Microsoft Azure PlayFab* [Corporation 2021] que permite coletar dados de forma personalizada da *gameplay* de cada jogador.

## 6. Apresentação do protótipo

O jogo está situado em um ambiente futurista casual, o jogador deve controlar o gatinho Quindim, que ao acordar, se vê em um ambiente desconhecido e sem seu dono. Quindim deve passar por diversos obstáculos do ambiente (na forma de plataformas e "inimigos") e desafios propostos pelos NPCs, como os mostrados nas Figuras 2(a) e 2(b).

Na Figura 3 são mostradas as versões *mobile* e *desktop*, lado a lado. Na versão *mobile*, a movimentação do personagem é feita através do *joystick* e utiliza-se o botão à direita para pular. Para falar com os NPCs é necessário apenas que Quindim encoste neles, mas para passar para a próxima fala ou interagir com os blocos nos desafios é necessário tocar nos locais adequados.

Na versão *desktop* a movimentação do personagem é feita através das teclas "A" para ir para esquerda, "D" para direita e barra de espaço para pular. A interação com NPCs e blocos é feita através do cursor do mouse.

Estão presentes dois tipos de desafios de pensamento computacional: desafios de "caminho" e desafios de "geometria". Há um total de oito desafios de caminhos, como o da Figura 2(b), onde o jogador deve criar, por meio dos blocos de função, um algoritmo que levará o triângulo amarelo até o quadrado brilhante. Os desafios de "caminho" são dados

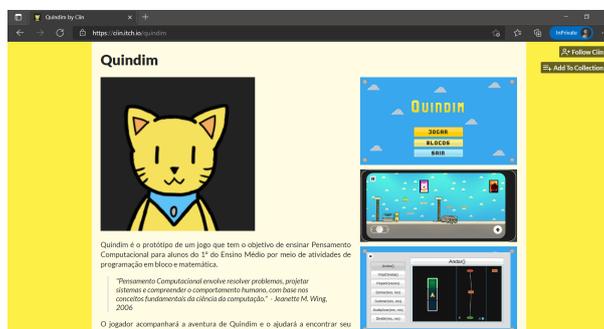


**Figura 3. Comparação entre versão Android e versão Desktop**

por dois personagens: Manchas e Botas, cada um com quatro desafios. Os desafios de "geometria", por sua vez, são dados por UwU, a coruja robótica que gerencia o elevador do prédio.

Ao passar pelos desafios de UwU, o jogador é então guiado pela personagem para a sala do chefe da empresa, onde se encontra o dono de Quindim e ao jogador é dada a conclusão da história.

O protótipo está disponível na loja de jogos virtual *itch.io* de forma não listada, portanto é necessário acessar o jogo através do *link*: <https://ciin.itch.io/quindim> onde estão disponíveis para *download* ambas as versões para *Windows* e para *Android*, como pode ser observado na Figura 4.



**Figura 4. Site do protótipo**

## 7. Potencial de aplicação do protótipo em sala de aula

Por ter sido desenvolvido tanto para versão móvel *android* quanto para versão *desktop*, o protótipo mostra vasto alcance para uso em sala e fora dela, já que os alunos podem jogar por meio de seus telefones celulares ou, caso possível, no laboratório de informática da escola. Além disso, o jogo é voltado para *gameplays* curtas, ou seja, é uma ferramenta complementar para o ensino de Pensamento Computacional e suas aplicações no cotidiano, ao trazer uma forma de o aluno experimentar por conta própria o que lhe foi ensinado.

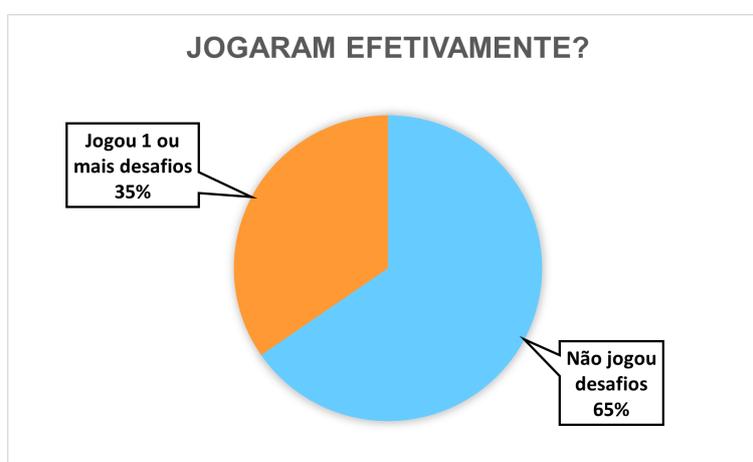
## 8. Resultados esperados e ou alcançados pela aplicação

O protótipo foi colocado para teste com alunos no período de 06 de junho de 2021 até 26 de julho de 2021. Nesse período ficaram disponíveis através do *Google Forms* dois

questionários: o preliminar e o de reação.

O questionário preliminar teve como foco diagnosticar o público, levantar questionamentos acerca dos hábitos e preferências dos participantes, entender um pouco das experiências do público com jogos educacionais relacionados a disciplina de matemática, programação e pensamento computacional. Obteve-se um total de 111 respostas válidas no período mencionado.

O questionário de reação deveria ser respondido após o uso do jogo. Esse, por sua vez obteve um total de 27 respostas. As respostas foram então analisadas e comparadas com os resultados da avaliação furtiva para confirmação de *gameplay*. A comparação, mostrada na figura 5, revelou que apenas 9 dos 27 participantes que responderam jogaram efetivamente o jogo, completando pelo menos um desafio.



**Figura 5. Comparação quantidade de respostas com quantidade de resultados da avaliação furtiva**

Imagina-se que os motivos possam ser o acúmulo de atividades (trabalhos, provas) do fim do semestre letivo, bem como o próprio período de férias dos alunos.

### 8.1. Avaliação Furtiva

A avaliação furtiva foi realizada utilizando o *Microsoft Azure PlayFab* [Corporation 2021], que permite a criação de credenciais únicas para cada jogador e a coleta de dados personalizados. No caso do protótipo são coletados os seguintes dados: fase atual e por desafio: nome do desafio, quantidade de tentativas, quantidade de erros de sintaxe, cada iteração única do código criado pelo usuário e tempo gasto no desafio.

A figura 6 mostra os dados coletados de um jogador em um desafio (exibido do lado esquerdo da imagem). Por meio desse resultado é possível ver que o jogador fez um total de cinco tentativas e gastou 5 minutos e 10 segundos no desafio. Também é possível perceber o modo como o código criado por ele progrediu a cada tentativa.

Além disso, utilizando o mesmo desafio como referência, é possível comparar os resultados dos diversos alunos, como na figura 7 e identificar e atender os alunos. Por exemplo, os jogadores 2 e 5 (da esquerda para a direita do gráfico), apresentaram tempo muito maior que os demais e podem receber orientação/reforço.

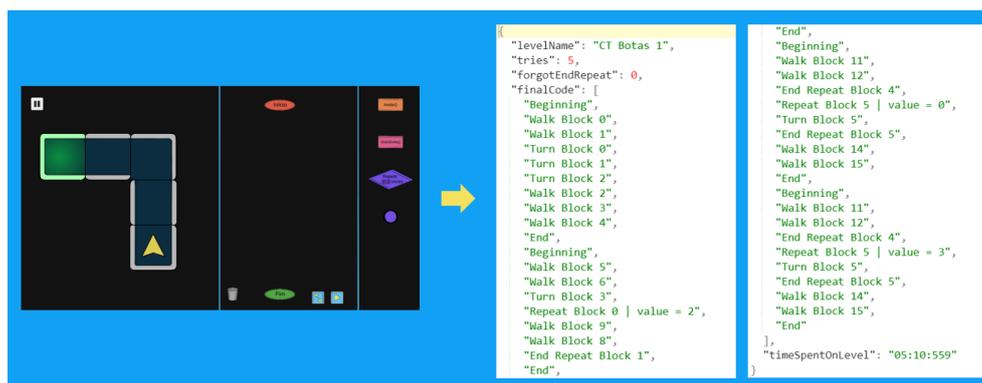


Figura 6. Resultado da Avaliação Furtiva de um Jogador - Desafio Botas 1

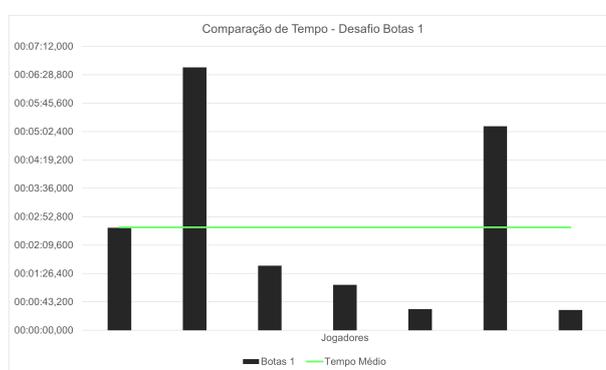


Figura 7. Resultado da Avaliação Furtiva de um Jogador - Desafio Botas 1

## 9. Considerações Finais dos Autores

O processo na descoberta de novos mundos dependere da motivação e da importância que cada um tem por novos desafios. O uso de jogos na educação pode ser algo que motive e desafie os alunos. O PC pode ser desenvolvido como mais uma estratégia à resolução de problemas, cabendo ao educador interagir com os alunos e observar onde estão as dificuldades apresentadas por estes. Neste aspecto, a avaliação furtiva pode ser de grande valia.

Espera-se que esse protótipo contribua nesse processo, considerando que ele adiciona possibilidades às práticas docentes já existentes, e coopere para o descoberta de novas práticas relevantes em tempos de transição para uma nova educação.

Para uma maior ubiquidade da utilização do protótipo de forma livre, para o público mais amplo, são projetos futuros: melhoria dos tutoriais, adição de novas fases para o jogo, criação de um banco de questões de matemática, inserção de um sistema de conquistas e prêmios, opção de criar novos blocos com funcionalidades personalizadas (como "VirarEsquerda()"), dar ao professor, caso o jogo esteja sendo utilizado em sala de aula, acesso ao resultados obtidos pela avaliação furtiva (visando permitir ao professor ajudar os alunos de forma individual) e a criação de uma versão para iOS, visto que hoje o mesmo tem apenas versões para *Android* e *Windows*.

## Referências

- Brackeys (2020). Brackeys. Disponível em: <https://www.youtube.com/user/Brackeys/>.
- CIEB (2018). Currículo de Referência em Tecnologia e Computação - CIEB. Disponível em: <https://curriculo.cieb.net.br/>.
- CocoCode (2021). Cococode. Disponível em: <https://www.youtube.com/c/CocoCode/>.
- CodeMonkey (2021). Codemonkey. Disponível em: <https://www.youtube.com/c/CodeMonkeyUnity/>.
- Corporation, M. (2021). Microsoft azure playfab. Disponível em: <https://playfab.com/>.
- Forquesato, L. (2018). Usando um jogo para ensinar pensamento computacional e avaliar o aprendizado.
- Foundation, K. (2021). Krita. Disponível em: <https://krita.org/en/>.
- GameLoot (2021). Gameloot. Disponível em: <https://www.youtube.com/c/GameLootDev/>.
- Kazimoglu, C., Bacon, L., Kiernan, M., and Mackinnon, L. (2012). Learning Programming at the Computational Thinking Level via Digital Game-Play Procedia Computer Science.
- Kenney (2020). Kenneygameassets1. Disponível em: <https://kenney.itch.io/kenney-game-assets-1>.
- MEC (2017). Base Nacional Comum Curricular (BNCC). Disponível em: [http://basenacionalcomum.mec.gov.br/images/BNCC\\_EI\\_EF\\_110518\\_versaofinal\\_site.pdf](http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf).
- Papert, S. (1986). *Constructionism: A New Opportunity for Elementary Science Education*. Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group, Massachusetts.
- Studio, M. V. (2021). Visual studio 2019. Disponível em: <https://visualstudio.microsoft.com/pt-br/vs/>.
- Team, T. A. (2021). Audacity. Disponível em: <https://www.audacityteam.org/>.
- Technologies, U. (2021). Unity. Disponível em: <https://unity.com/pt>.
- Turchi, T., Fogli, D., and Malizia, A. (2019). Fostering computational thinking through collaborative game-based learning. *Multimedia Tools and Applications*, 78:13649–13673.
- Valente, J., Freire, F., Arantes, F., d’Abreu, J., Amiel, T., and Baranauskas, M. C. (2017). Alan turing tinha pensamento computacional? reflexões sobre um campo em construção. In *Tecnologias, Sociedade e Conhecimento*, pages 7 – 22.
- Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3):33–35.