

Run Code: Um Jogo Mobile Auxiliador no Processo de Ensino e Aprendizagem em Lógica de Programação

Caio Max G. F. Trindade¹, Tiago do Carmo Nogueira², Naidson Clayr S. Ferreira¹

¹Instituto Federal Baiano – Campus Guanambi (IFBAIANO)
Caixa Postal 09 – Distrito de Ceraíma – 46.430-000 – Guanambi – BA – Brazil

²Instituto Federal Baiano – Campus Bom Jesus da Lapa
BR 349, Km 14 – Zona Rural – Bom Jesus da Lapa – Bahia, CEP: 47600-000.

caio.max3825@gmail.com, tiago.nogueira@ifbaiano.edu.br

naidson.ferreira@gmail.com

Abstract. *The present work proposes the development of a serious game mobile, in which it would use the basic elements of programming. For the development of the game, Unity Engine was used as the main development tool, which was divided into 8 stages, starting with the preparation of the environment, proceeding to the creation of the script where the scenario that the game set in described, as well as each character and their role in the game, finally elaborating the characters' dialogues. After the script developed, the game's assets graphics were created, for them the software Aseprite was used, making all the modifications to pre-existing assets and creating others, based on the elements defined in the roadmap. With the assets developed, they were used to create the game components in Unity, which in the next stage had scripts encoded in C # to make game work. With all the elements developed, the game phases created and performance tests performed on them. During the tests it was possible to verify that all devices presented a rate above 60 frames per second, maximum supported by the devices. Finally, the whole project compiled in two versions, the main one for Android and the other one for Windows.*

Resumo. *O presente trabalho propõe o desenvolvimento de um serious game mobile, no qual utilizaria os elementos básicos de programação. Para o desenvolvimento do jogo foi utilizado como principal ferramenta de desenvolvimento o Unity Engine, que foi dividido em 8 etapas, iniciando com a preparação do ambiente, passando para a criação do roteiro onde o cenário que o jogo se passa é descrito, conforme assim como cada personagem e seu papel no jogo, por fim elaborando os diálogos dos personagens. Após o desenvolvimento do script, foram criados os gráficos dos assets do jogo, para eles foi utilizado o software Aseprite, fazendo todas as modificações nos assets pré-existentes e criando outros, com base nos elementos definidos no roteiro. Com os assets desenvolvidos, eles foram usados para criar os componentes do jogo em Unity, que na próxima etapa tiveram scripts codificados em C# para fazer o jogo funcionar. Com todos os elementos desenvolvidos, as fases do jogo criadas e os testes de desempenho realizados nas mesmas. Durante os testes foi possível verificar que todos os aparelhos apresentaram taxa acima de 60 frames por segundo, máximo suportado pelos aparelhos. Por fim, todo o projeto compilado em duas versões, a principal para Android e outra para Windows.*

1. Introdução

Atualmente a Tecnologia da Informação tornou-se parte importante da infraestrutura mundial, assim surgiu a necessidade de educar os jovens digitalmente, porém a educação digital básica empregada inicialmente, mostrou-se ineficiente, treinando os jovens apenas como usuários de ferramentas [García-Peñalvo 2018].

Neste sentido, a tecnologia avança em um ritmo que torna inviável treinar uma pessoa para desempenhar um papel ou usar uma ferramenta, a qual logo poderá ou irá tornar-se obsoleta. Com isso, surgiu outra necessidade, a do jovem saber como são as regras na qual essas ferramentas são construídas e baseadas. Dessa forma, o pensamento computacional surgiu como um novo paradigma e a programação se estabelece como um resolutor de problemas [García-Peñalvo 2018].

Esse processo de aprendizado vem encontrando uma barreira, a dificuldade dos estudantes em compreender o raciocínio lógico e conceitos utilizados na programação. Empecilho este que fora construído com elementos básicos a outros processos de aprendizado, como a vertente matemática empregada na computação.

Contudo, fatores externos surgem evidenciando essa dificuldade. Neste aspecto, [Figueiredo and García-Peñalvo 2018], incluem entre esses fatores a falta de motivação, o fracasso e a frustração dos alunos.

Nesta ótica e à procura de um elemento que engaje o estudante, este projeto optou por seguir explorando a utilização dos denominados *serious games*, criando assim uma abordagem com a mesma.

O *game* é uma forma de entretenimento interativa, a qual em muitos são propostos desafios, no qual o jogador deve realizá-los de modo a obter sucesso no seu objetivo, assim para esse projeto, os jogos estabelecem dois aspectos importantes, o primeiro é a própria interação do jogador e o segundo a tentativa e erro.

Em jogos que não propõe punir severamente o jogador pelos seus erros, diferente da realidade, falhar não representa graves consequências ao jogador, visto que ele pode recomençar, e utilizar do seu erro para aprender o que deve ser feito.

Utilizando dos aspectos mencionados, cria-se um cenário onde, ao experimentar essa experiência, o jogador é convidado a interagir com um mundo, que o ensina realizar determinado objetivo, podendo utilizar suas falhas para alcançar o sucesso. Assim seguindo a linha de raciocínio apresentada chega-se aos jogos sérios.

Utilizado em 1970 por *Clark Abt*, o termo *serious games*, surge empregando a forma de entretenimento proporcionado pelos jogos, para o ensino e aprendizado. Os jogos sérios transportam as formas de ensinar e aprender, a um jogo, sendo esse método utilizado por diversas áreas, desde a científica a militar.

Dessa forma a aplicação dos *serious games* traz os conceitos normalmente aplicados em determinadas situações e os expõem sobre uma nova ótica ou contexto, possibilitando assim, um cenário onde o receptor do conteúdo, correlacione sua teoria com a prática.

Assim, o objetivo deste trabalho, além de desenvolver um *game*, no qual apresenta os conceitos envolvidos na lógica de programação, objetiva-se também, realizar um teste

de performance da mesma.

A metodologia encontra-se dividida em três fases onde, na primeira fase, são apontados como funcionará o jogo. Na segunda fase é descrito o escopo do jogo, assim como as ferramentas utilizadas para o mesmo, por fim, na terceira fase, é relatado todo o processo de desenvolvimento.

2. Trabalhos relacionados

Nesse capítulo é apresentado trabalhos cujos temas são relacionados ou parecidos com este projeto. Neles pode se observar como os jogos sérios já são empregados para o aprendizado de lógica e linguagem de programação, assim como diversas outras áreas do conhecimento.

Nesse sentido, [Mitamura et al. 2012] aponta como os “jogos sérios” vem recebendo atenção nos últimos anos, com o conceito que pode-se aprender enquanto se diverte. Após relatarem como pode não ser prazeroso o processo tradicional de ensino de programação, os autores propõem e avaliam jogos sérios para o ensino da programação.

Não obstante, para [Alsubaie et al. 2018], as crianças vêm gastando o tempo que deveria ser investido em estudo com atividades de lazer, segundo os mesmos, isso se dá, pois, os processos de aprendizado são lentos, rígidos e não permitem a criatividade.

Assim afirmando, que jogos educacionais podem ser um truque para as crianças se interessarem pela educação. Dessa forma, uma abordagem com um jogo educativo pode surtir efeitos nas crianças, principalmente com a expansão dos *smartphones*.

[Pellas and Vosinakis 2018] propuseram uma abordagem para medir como a interfaces de um jogo educacional e elementos de mundos virtuais em 3D, combinados com uma linguagem de programação visual, impactam no aprendizado de estudantes do colegial. Assim, a abordagem contou com dois grupos de 25 estudantes. Os resultados demonstram que, o grupo que utilizou da junção dos três elementos, conseguiu fazer um sistema completo, aplicando conceitos de programação corretamente mais vezes.

[Martin-Dorta et al. 2014] também utilizaram um jogo 3D chamado de *Virtual Blocks*, esse jogo permite a construção de modelos com cubos em um ambiente 3D *mobile*. Procurando desenvolver a capacidade espacial, no curso de engenharia, o estudo contou com 26 calouros e apresentou resultados expressivos, de impacto mensurável na capacidade espacial dos estudantes, estimulando-os e facilitando a sua aplicabilidade.

Procurando trazer o interesse do aluno às matérias de programação tanto [Malliarakis et al. 2013] quanto [Tsalikidis and Pavlidis 2016] utilizam elementos *online* em um jogo nos moldes de um *Massively Multiplayer Online Role-Playing Game*, abreviado como MMORPG, é um gênero de *game* onde uma quantidade massiva de jogadores se juntam *online* para interagir entre se e interpretar papéis disponíveis naquele mundo virtual. [Malliarakis et al. 2013] desenvolveu um MMORPG voltado para o ensino da programação introdutória, [Tsalikidis and Pavlidis 2016] criou um MMORPG onde os jogadores podem aprender ou treinar seu conhecimento em *javascript*.

Em um estudo realizado na Espanha, [González et al. 2013] buscava através de videogames incentivar alunos a aprenderem programação para sistemas embarcados, visto que muitos alunos veem elementos dos sistemas embarcados com estranheza e diferentes

dos computadores pessoais. Com resultados muito satisfatórios, o número de matrículas aumentou e houve uma melhoria na média final dos alunos.

Em seu trabalho [Keengwe and Bhargava 2014] afirma que as tecnologias *mobile* tem o potencial de transformar a educação. Após avaliar os benefícios pedagógicos dessas tecnologias como ferramentas educacionais, os resultados obtidos relatam que se a tecnologia for usada de forma adequada, projetada e desenvolvida em um contexto, pode ter enorme impacto positivo na sociedade.

Nesta perspectiva, jogos educacionais mobiles estão surgindo cada vez mais e com para abordagens em mais áreas, [Chang and Yang 2016] desenvolveram um jogo para auxiliar o aprendizado da matemática no ensino fundamental, visto que abordagens tradicionais podem ser desinteressantes para os alunos. Constatou-se que os alunos conseguiram estudar de forma autônoma e aumentou seu interesse e motivação por aprender matemática.

Do mesmo modo [Elaish et al. 2019] obtiveram resultados parecidos quando desenvolveram um jogo sério mobile, porém, o seu *game* auxilia estudantes árabes a aprender e praticar inglês.

[Lotfi and Mohammed 2018] propuseram um jogo sério mobile para o ensino de programação orientada a objetos. No entanto, não conseguiram medir sua eficácia com a ferramenta de avaliação inclusa no game. [Wong et al. 2017] propôs um game com proposta parecida, mas dessa vez seria um *RPG* ensinado os paradigmas da orientação a objetos, e parte dos participantes do teste do *RPG* concordaram que o jogo é uma eficiente ferramenta no que se propõem.

[Jordine et al. 2014] criaram um projeto no qual consistia em criar uma abordagem para ensinar conceitos de programação em java com dispositivos móveis, parte dessa abordagem constituiu em desenvolver um protótipo de jogo mobile.

Dessa mesma forma [Sierra et al. 2016] também desenvolveu um *game* para o auxílio no aprendizado da linguagem java, com os resultados obtidos do questionário aplicado aos participantes, o jogo mostrou ser fácil de usar e eficiente em reforçar os conteúdos da disciplina de programação, mesmo com usuários com conhecimento mínimo.

3. Metodologia

O *game* desenvolvido possui intuito de mostrar aos jogadores como ocorre o fluxo de leitura em um código de programação estruturado, como os elementos do código integram e exemplificam de forma lúdica o funcionamento de um algoritmo e um estrutura de repetição. Para tal, na seção 3.1 é apresentado o conceito do jogo, na seção 3.2 será mostrado o escopo do projeto, em 3.3 é apresentado a introdução ao processo do desenvolvimento.

3.1. Conceito

Para cumprir seu objetivo, o aplicativo designa ao jogador ir do ponto A, ao ponto B, desviando de obstáculos no meio do caminho, tendo disponíveis apenas algumas funções, as quais foram apresentadas por seu guia. O jogador terá que traçar qual será o caminho a

ser percorrido e quais ações disponíveis ele deverá utilizar. Assim o jogador selecionará previamente quais ações deverão ser executadas e sua respectiva ordem.

Uma vez finalizado o jogo terá uma sequência finita de passos lógicos, a qual será utilizada para executar os comandos do personagem, o jogador poderá ver se o objetivo foi concluído. A mecânica principal de jogo visa fazer o jogador construir um algoritmo, para que ele possa cumprir seu objetivo nas fases. Ao decorrer das fases serão apresentadas as mecânicas do jogo, as quais deverão ser utilizadas para compor o algoritmo.

3.1.1. Aplicação do conceito no jogo sério

O jogo abordará em suas fases, algoritmos e estrutura de repetição, de modo a introduzir esses conceitos de forma incremental, o jogo possui 10 fases, sendo as 5 primeiras focadas em fazer o jogador aprender o conceito de algoritmo, que será utilizado ao decorrer de todo o game.

Na primeira fase é explicado ao jogador, que para se movimentar pela fase até à saída, ele deverá definir uma sequência de instruções lógicas. Durante a primeira fase a única instrução disponível é a de avançar para frente, mais instruções serão inseridas mais a frente. As instruções são inseridas gradualmente ao decorrer do jogo, o objetivo das primeiras 5 fases, é fazer o jogador se acostumar com conceito de execução de uma sequência lógica de intuições, assim a princípio existem, poucos elementos para interagir.

Durante a segunda e terceira fase, é inserido respectivamente as instruções responsáveis girar o personagem e fale-ló pular, assim durante a apresentação de cada, é explicado seu funcionamento, e como ela deve ser usada na sequência lógica.

A quarta fase não apresenta uma nova instrução, porém, é apresentado um novo elemento no cenário, assim é explicado o seu funcionamento ao jogador e como o algoritmo gerado interage com esse elemento. À medida que mais elementos são inseridos procura-se gradualmente desafiar o jogador, a utilizar o raciocínio lógico, para integrar as instruções à sua disposição, de modo a montar uma sequência lógica que resolva o desafio da fase.

Na quinta fase, o jogo apresenta uma nova instrução, a de atacar os oponentes no cenário, todas as instruções disponibilizadas até aqui, não representam explicitamente um conceito relacionado a programação, porém, elas trabalham juntas para mostrar ao jogador o funcionamento de um algoritmo, além de exercitar o raciocínio lógico.

A sexta fase do jogo, apresenta a última instrução disponibilizada ao jogador, ela traz ao game a representação de uma estrutura de repetição, no jogo ela é utilizada, para que o jogador possa repetir uma sequência de movimento até chegar no seu objetivo.

No jogo, o jogador só pode inserir por padrão 27 instruções, assim repetir trechos grandes várias vezes se torna inviável, o modo que a fase está construída permite ao jogador utilizar a estrutura de repetição, a partir de certa posição no cenário. Com as informações da estrutura de repetição apresentada ao jogador, o jogo vai colocando trechos onde esta pode ser utilizada, assim a partir da sexta fase até o final, o jogo vai exercitando e ensinado o jogador a utilizar o *loop*, nas sequências de instrução que serão criadas.

Desta forma, da sexta fase até a última, vai sendo trabalhado nos diálogos, o funcionamento e as características da estrutura de repetição disponibilizada, além de ir inserido mais elementos ao cenário, exigindo do jogador utilizar gradualmente mais instruções, para gerar seu algoritmo.

Com o conceito do jogo idealizado, o próximo passo é elaborar o escopo dele no *GDD (Game Design Document)*, nele será definido, todas as características que devem ser apresentadas no produto final, além de elaborar as metas e etapas para o desenvolvimento.

3.2. Game design document

O desenvolvimento de um jogo demanda de várias etapas, estas demandam de recursos e tempo, cada detalhe deve ser considerado e analisado antes de seu desenvolvimento. Nesta seção será detalhado o planejamento e definição do escopo do jogo no *GDD*.

No *GDD*, todas as informações acerca do jogo e seu desenvolvimento são reunidas, ele descreve o jogo, desde sua sinopse, características, a definições gerais, como plataformas utilizadas, gênero, público alvo, entre outras.

Além de detalhar o jogo, o *GDD* descreve as etapas do processo de desenvolvimento, a meta de cada etapa e as atividades envolvidas para se alcançar essa meta.

Neste projeto o *GDD* é adaptado para uma versão voltada há um projeto desenvolvido por uma pessoa apenas, baseado no estudo de caso realizado por [Motta and Junior 2013]. A presente seção vem com intuito de mostrar os principais pontos do *GDD*, em relação ao desenvolvimento, iniciando pelas definições gerais do jogo.

3.2.1. Definições gerais

Na elaboração do jogo, é preciso ter bem claro seus aspectos mais básicos, como seu gênero, as plataformas no qual será lançado, seu público alvo, entre outros. Analisando o conceito inicial do jogo, pode se chegar as seguintes definições:

- **Gênero:** *Puzzle Game*;
- **Plataformas:** *Android* e *Windows*;
- **Quantidade de níveis:** 10 jogáveis + 1 de história;
- **Quantidade de vilões/inimigos:** Um inimigo que se repete;
- **Público alvo:** Entre 15 e 20 anos.

Os itens definidos, são apenas descrições genéricas, que serviram de base com o conceito, para a elaboração das características mais específicas.

3.2.2. Características

Diferente das definições gerais, as características definidas, aqui serviram como indicativo, do que pode ser encontrado no produto final.

Ao definir as características é de extrema importância que elas não sejam inalcançáveis e caso surjam novas durante o desenvolvimento, seja verificado se é possível incluí-las na atual versão.

Para o jogo foi definido as seguintes características:

- O jogo deverá contar com uma ambientação medieval;
- A movimentação do personagem deve ser o maior desafio do jogo;
- O jogador terá múltiplas ações para interagir e movimentar no cenário;
- O cenário deverá conter obstáculos que promovam desafios ao jogador;
- Deverá exigir do jogador o uso do raciocínio lógico e atenção no funcionamento de cada elemento presente na fase;
- Deverá possuir uma versão para *Android* e outra para *Windows*;
- Em dispositivos *Android*, com processador *Quad Core* e *2Gb* de *RAM*, sua performance precisa atingir acima de *30 FPS*;

As características definidas mostram o que deve se esperar da versão do jogo finalizada, para que eles realmente estejam no jogo é preciso definir como serão desenvolvidas e quais serão as etapas para seu desenvolvimento. O próximo tópico abordado, trará as definições das etapas para o desenvolvimento do game, suas metas e atividades.

3.2.3. Metas e atividades

Para se alcançar o objetivo final do projeto, e desenvolver o jogo de acordo ao planejado, é necessário ter metas que sejam alcançadas durante o desenvolvimento, essas indicam estágios do desenvolvimento do jogo.

Cada meta é composta de atividades que devem ser realizadas, para a realização do objetivo almejado. O desenvolvimento do jogo é composto de 8 metas, quando todas as atividades que compõem uma meta, são concluídas com êxito, a meta da etapa é encerrada, e iniciada a próxima etapa, com seus objetivos e atividades.

Cada etapa, tem definida a quantidade de tempo que deve ser alocada para sua realização, porém, atrasos durante as realizações das atividades podem ocorrer, sendo necessário mais tempo para a realização desta. Todas as etapas, com suas metas e atividade podem ser encontradas a seguir:

- **1ª Etapa:** Preparar ambiente de desenvolvimento - 3 dias;
- **2ª Etapa:** Elaborar roteiro - 8 dias;
- **3ª Etapa:** Criar e modificar *assets* gráficos - 9 dias;
- **4ª Etapa:** Criar os *GameObjects* no *Unity Editor* - 10 dias;
- **5ª Etapa:** Desenvolver a codificação do jogo - 14 dias;
- **6ª Etapa:** Criar as fases - 7 dias;
- **7ª Etapa:** Realizar testes - 7 dias;
- **8ª Etapa:** Lançar os primeiros *releases* - 1 dia.

Com todas as atividades elaboradas, as ferramentas necessárias para o desenvolvimento, encontram-se citadas. O próximo capítulo abordará as ferramentas utilizadas no desenvolvimento e suas funcionalidades.

3.3. Ferramentas utilizadas

Para o desenvolvimento da aplicação foi utilizado a *Unity*, uma *engine* que atualmente está na sua quinta versão, um *software* capaz de facilitar o processo de desenvolvimento de um jogo e afins.

A *engine* inclui o motor para a renderização de modelos 3D ou 2D, motor de física, permitindo a simulação de leis da física. A codificação do jogo foi realizada utilizando-se da linguagem C# (*Microsoft .NET 4.6*).

Para a codificação do jogo foi utilizado o editor de códigos, chamado *Visual Studio Code*, essa ferramenta gratuita permite o suporte a diversas linguagens. O *Visual Studio Code* utiliza extensões, para ampliar seu suporte a diferentes tecnologias.

Por fim, para os elementos no jogo são denominados de *Assets*, eles são desde personagens e cenários à inteligência artificial de um *game*.

Assets são elementos essenciais em um *game*, garantindo a imersão e abstração do conteúdo passado por ali. Para os *assets* gráficos que serão utilizados no desenvolvimento, alguns utilizam bibliotecas online, onde podem ser encontradas tanto de forma gratuita quanto pagas.

Para o desenvolvimento de *assets* próprios para o jogo, será utilizado o *Aseprite*, este é uma ferramenta paga, o qual sua aquisição por meio da *Steam* será o único custo monetário com software do projeto.

Os projetos gerados pelo *Unity* podem ser exportados para diversas plataformas, sejam *Desktop* ou *mobile*. Para o desenvolvimento desse *app* ele será exportado para *Android* e para o *Windows*.

Com todo o processo de elaboração do projeto finalizado, pode ser iniciado, as etapas de desenvolvimento do jogo, iniciando pela preparação das ferramentas apresentadas.

4. Resultados e Discussões

Como resultado obtido do processo de desenvolvimento, tem se um jogo sério que apresenta em suas dez primeiras fases o conceito de algoritmo, além de introduzir estruturas de repetição.

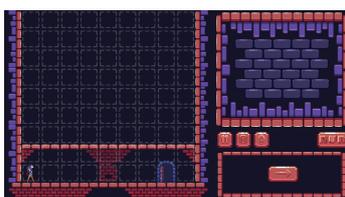


Figura 1. Primeira fase



Figura 2. Caixa de diálogos

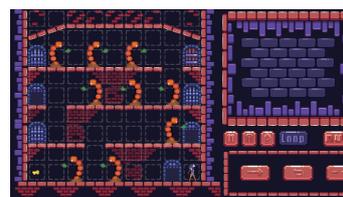


Figura 3. Última fase

A figura 1, representa a primeira fase, essa introduz pela primeira vez o conceito de algoritmo, e conceitua e ambienta o jogador na situação que se encontra. Os diálogos representados pela figura 2, exercem o papel de apresentar para o jogador a história do jogo, a qual avança pela interação dos dois personagens presentes, e transmite através dela os conceitos a serem passados. Na figura 3 apresenta a última fase do jogo, pelo fato do jogo ir introduzindo elementos gradualmente, ela conseqüentemente se torna a fase mais complexa do *game*, nela o jogador deverá usar seu aprendizado sobre algoritmos e estrutura de repetição, além dos elementos específicos do game.

Tabela 1. Média fps em cada fase do jogo

	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5	Fase 6	Fase 7	Fase 8	Fase 9	Fase 10	Média	Mediana	Desvio Padrão
Galaxy S5 mini	62	62	62	62	62	61	61	61	61	61	61,5	0,27	0,52
Redmi 6A	62	63	62	61	61	61	62	62	61	62	61,7	0,45	0,67
Galaxy A10	60	60	60	60	60	60	60	60	60	60	60	0	0

Na tabela 1 é apresentado os dados obtidos durante o teste de desempenho realizado em 3 dispositivos diferentes. No teste de desempenho foi observado a média de quadros gerados por segundo, durante a execução de cada fase.

5. Conclusões

O projeto desenvolvido neste trabalho veio com a proposta de desenvolver um jogo sério, de modo a apresentar de forma lúdica, alguns conceitos básicos relacionados a programação, visto que esses podem tornar-se confusos inicialmente, sendo estes os conceitos de raciocínio lógico e estruturas de repetição.

Os objetivos do projeto foram alcançados ao utilizar da *Unity Engine* para criar um jogo educacional para as plataformas *Android e Windows*. Com os resultados obtidos pode-se evidenciar um jogo que utiliza de sua mecânica baseada em raciocínio lógico aliado com sua história, para apresentar alguns conceitos básicos da programação.

Durante todo o processo de desenvolvimento do *game*, foi possível observar, as principais etapas do desenvolvimento de um jogo na *Unity*, desde sua concepção a sua compilação, mostrando um processo que envolve múltiplas áreas, sendo a programação apenas uma delas.

Conclui-se assim que o presente trabalho, o qual resulta em um jogo sério, que coloca o jogador com parte de uma história a qual ele precisa participar ativamente na busca pela resolução do problema, e compreender que cada elemento apresentado a ele faz parte de um todo, assim como na programação.

Ao final o jogo resulta em uma transposição de conceitos básicos, em um contexto lúdico, podendo atuar como um auxílio e suporte para compreensão de conhecimento pré-estabelecidos ou uma simples introdução a conceitos da programação, que devem ser aprofundados de outras formas.

Referências

- Alsubaie, A., Alaithan, M., Boubaid, M., and Zaman, N. (2018). Making learning fun: Educational concepts & logics through game. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pages 454–459. IEEE.
- Chang, R.-C. and Yang, C.-Y. (2016). Developing a mobile app for game-based learning in middle school mathematics course. In *2016 International Conference on Applied System Innovation (ICASI)*, pages 1–2. IEEE.
- Elaish, M. M., Ghani, N. A., Shuib, L., and Al-Haiqi, A. (2019). Development of a mobile game application to boost students' motivation in learning english vocabulary. *IEEE Access*, 7:13326–13337.
- Figueiredo, J. and García-Peñalvo, F. J. (2018). Building skills in introductory programming. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 46–50. ACM.

- García-Peñalvo, F. J. (2018). Editorial computational thinking. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 13(1):17–19.
- González, J., Pomares, H., Damas, M., García-Sánchez, P., Rodríguez-Alvarez, M., and Palomares, J. M. (2013). The use of video-gaming devices as a motivation for learning embedded systems programming. *IEEE Transactions on Education*, 56(2):199–207.
- Jordine, T., Liang, Y., and Ihler, E. (2014). A mobile-device based serious gaming approach for teaching and learning java programming. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pages 1–5. IEEE.
- Keengwe, J. and Bhargava, M. (2014). Mobile learning and integration of mobile technologies in education. *Education and Information Technologies*, 19(4):737–746.
- Lotfi, E. and Mohammed, B. (2018). Teaching object oriented programming concepts through a mobile serious game. In *Proceedings of the 3rd International Conference on Smart City Applications*, page 74. ACM.
- Malliarakis, C., Satratzemi, M., and Xinogalos, S. (2013). Towards a new massive multi-player online role playing game for introductory programming. In *Proceedings of the 6th Balkan Conference in Informatics*, pages 156–163. ACM.
- Martin-Dorta, N., Sanchez-Berriel, I., Bravo, M., Hernandez, J., Saorin, J. L., and Contero, M. (2014). Virtual blocks: a serious game for spatial ability improvement on mobile devices. *Multimedia Tools and Applications*, 73(3):1575–1595.
- Mitamura, T., Suzuki, Y., and Oohori, T. (2012). Serious games for learning programming languages. In *2012 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 1812–1817. IEEE.
- Motta, R. L. and Junior, J. T. (2013). Short game design document (sgdd). *Proceedings of the XII SBGames*, pages 16–18.
- Pellas, N. and Vosinakis, S. (2018). The effect of simulation games on learning computer programming: A comparative study on high school students’ learning performance by assessing computational problem-solving strategies. *Education and Information Technologies*, 23:2423–2452.
- Sierra, A., Ariza, T., Fernández-Jiménez, F., Muñoz-Calle, J., Molina, A., and Martín-Rodríguez, Á. (2016). Educational resource based on games for the reinforcement of engineering learning programming in mobile devices. In *2016 Technologies Applied to Electronics Teaching (TAEE)*, pages 1–6. IEEE.
- Tsalikidis, K. and Pavlidis, G. (2016). jlegends: Online game to train programming skills. In *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–6. IEEE.
- Wong, Y. S., Hayati, M., Yatim, M., and Hoe, T. W. (2017). A propriety game based learning mobile game to learn object-oriented programming—odyssey of phoenix. In *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 426–431. IEEE.