

Jogo de Plataforma no Auxílio Dinâmico ao Ensino de Orientação a Objetos - Manuel Adventures

Alexandre Vitor S. Braga¹, Igor C. Rodrigues¹,
Marcelo Ian R. Menezes¹, José Ronaldo Mouro¹

¹ Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora (UFJF) – Juiz de Fora, MG – Brazil

{alexandre.braga, igorcorrea.rodrigues}@estudante.ufjf.br

{marcelo.menezes, 12068068850}@estudante.ufjf.br

Abstract. *Since the Object-Oriented Programming subject is an essential discipline for Computer Science courses and related areas, and also one of the subjects with the highest failure rate. This article presents Manuel Adventures, an approach to support the teaching and learning of Object-oriented Programming, in a playful and interactive way using serious game as a resource. Its implementation was done using HTML, CSS and Typescript programming language, without any game engines. Manuel Adventures' objective is to be more dynamic in comparison to its counterparts, being a way to attract a much bigger audience of students. The project prototype was tested internally by the development team, points were the game could be improved were found, and suggestions were made.*

Resumo. *Visto que a disciplina de Programação Orientada a Objetos é uma disciplina essencial para os cursos de Ciência da Computação e áreas afins, além de ser uma das disciplinas com maior índice de reprovação. Este artigo apresenta o Manuel Adventures, que tem como objetivo auxiliar no ensino e aprendizagem de Orientação a Objetos, de maneira lúdica e interativa usando como recurso jogos sérios. A sua implementação foi realizada em HTML, CSS e Typescript, sem o auxílio de game engines. O objetivo de Manuel Adventures é ser mais dinâmico que os jogos sérios convencionais a fim de interessar um público maior de estudantes. O projeto foi testado internamente pela equipe de desenvolvimento em sua versão protótipo, foram encontrados pontos a serem melhorados e sugestões realizadas.*

1. Introdução

A Disciplina de Programação Orientada a Objetos (POO) é considerada uma das mais difíceis pelos estudantes de graduação, apoiado pelo fato de ter uma das maiores taxas de evasão e reprovação no curso de Ciência da Computação, Engenharia Computacional e outros cursos correlatos. As dificuldades de aprendizagem encontradas pelos alunos quando ensinados pelo método tradicional, ou seja, com o professor apresentando a matéria no quadro ou através de *slides*, expondo exemplos relacionados e exercícios, normalmente relacionam-se com a falta de abstração dos estudantes para compreender suficientemente a lógica por trás dos programas desenvolvidos [Eliane M. Duarte de Barros 2013], a falta de conhecimento sobre a linguagem usada na

disciplina e sobre o ambiente de desenvolvimento integrado. Diante de tantas dificuldades, os estudantes se sentem desmotivados a aprender. Uma forma de facilitar o aprendizado dessa matéria seria a exposição de conceitos de programação estruturada, antes do estudo de Orientação a Objetos (OO). Esse paradigma de programação traz conceitos de estruturação sequencial, condicional e de repetição, deixando o estudante melhor preparado para estudar os conceitos mais abstratos de OO.

Ademais nos últimos anos está em voga a discussão acerca do pensamento computacional, a habilidade de utilizar conceitos e técnicas de áreas correlatas a Ciência da Computação em diversos campos [Ortiz and Pereira 2021]. Assim, faz-se necessário buscar maneiras de tornar o aluno mais ativo no processo de aprendizagem, no sentido de adoção de novas tecnologias e metodologias conjuntamente ao processo de ensino. Uma forma de fazer isso é através do uso de jogos, que podem trazer elementos de aprendizagem. O controle do progresso por parte do aluno que deve se manter constantemente desafiado e motivado, além de mecânicas assimétricas para diversificar a experiência são possíveis exemplos [Mattar and Nesteriuk 2016]. Logo, manter o interesse do aluno num jogo digital enquanto ensinamentos são transmitidos é um desafio que desperta o interesse da comunidade acadêmica.

A utilização de jogos como auxílio no aprendizado foram abordados em outros trabalhos, por exemplo em [Wong et al. 2018], [Mombach et al. 2018], [Lotfi and Bouhorma 2018]. Nesses projetos, a relação entre aluno e aprendizado foi tratada cuidadosamente para facilitar o entendimento do conteúdo analisando a influência das mídias digitais têm nos estudantes de OO. Especificamente, os jogos possuem alta acessibilidade por entre os jovens estudantes de cursos da computação, portanto, vêm atraindo interesse da comunidade acadêmica.

Usando desses princípios, esse artigo apresenta *Manuel Adventures*, um jogo sério para o auxílio da aprendizagem de Programação Orientada a Objetos (POO), disciplina básica e essencial para os cursos de Ciência da Computação, Engenharia Computacional, Sistemas de Informação e áreas afins.

O restante do trabalho está assim estruturado: na Seção 2 é feita uma breve abordagem acerca do uso de jogos para auxiliar no ensino de Orientação a Objetos; a Seção 3 descreve a forma como o jogo foi implementado e um pouco sobre seu conteúdo, enquanto a Seção 4 apresenta alguns trabalhos relacionados ao tema sobre o uso de jogos no ensino, bem como as metodologias e conceitos; a Seção 5 apresenta-se a análise comparativa entre a literatura e o projeto desenvolvido, e, por fim, a Seção 6 traz as conclusões do trabalho e propostas de trabalhos futuros.

2. Uso de Jogos no Ensino e Aprendizagem de Orientação a Objetos

O ato de jogar pode ser definido como uma atividade voluntária e livre que traz um sentimento de diversão e euforia. O jogador, quando se sente forçado a isso faz com que as características básicas do jogo se percam. A diversão em si só ocorre quando há um desejo por jogar, um sentimento de fuga da realidade e da rotina [Caillois and Barash 1961]. Nesse contexto, a noção de *game science* surge com o intuito de estabelecer a eficácia dos jogos na aprendizagem. Tal tarefa é complicada de ser definida, sendo mantida dentro de uma área mais dinâmica e robusta, na qual busca o entendimento de como é feito o processo de aquisição de conhecimento. Porém, quando vista de uma perspectiva educacio-

nal, a mesma cruza áreas multidisciplinares redesenhando questões acerca do aprendizado gamificado.

Contribuindo para esse cenário, a metodologia de aplicação da disciplina é tido como um agravante ao dificultar a compreensão dos alunos. O modo tradicional de ensino, onde uma figura central explica o conteúdo numa aula expositiva com eventuais exercícios e testes padronizados¹ contribui para o desinteresse dos estudantes. Nesse espectro, faz-se necessário o desenvolvimento de novas metodologias de ensino, que, para o século XXI, é possível o uso de recursos tecnológicos aos processos educacionais [Cleiton José Richter 2018]. Com a expansão das tecnologias digitais, o uso de jogos no ensino e aprendizagem de Orientação a Objetos torna-se um recurso extra para auxiliar os estudantes a compreender os conceitos básicos da matéria ao abstrair o conteúdo e apresentá-lo de forma interativa e divertida.

Jogos como recursos lúdicos são capazes de prover uma experiência satisfatória ao aluno, tanto no quesito diversão quanto no aprendizado, e, portanto, são uma forma do educador agregar métodos digitais como um complemento ao ensino direto [de Freitas 2018]. Nesse enquadramento, o uso de jogos para o auxílio no ensino da disciplina de Orientação a Objetos (OO) vem a ser relevante no contexto atual. A disciplina de OO por si só é considerada um ponto de entrada a muitas outras do curso de Ciência da Computação, e, como a disciplina é muito teórica, o seu ensino torna-se um desafio. Nesse contexto busca-se uma maneira de adequá-la ao aprendizado moderno, como forma alternativa de atrair possíveis interessados na área ao mesmo tempo que ajuda aqueles alunos com dificuldades.

O impacto dos jogos no aprendizado de Programação Orientada a Objetos se mostram positivas [Cleiton José Richter 2018] [Rafael Lucas da Costa 2021]. Utilizar um método amigável para o ensino do tema provê um maior engajamento e interesse dos alunos, consequentemente facilita a fixação do seu conteúdo aos estudantes. Nesse espectro, a utilização dessa metodologia é viável através de um planejamento didático e pedagógico consistentes utilizando ferramentas computacionais adequadas. Por outro lado, o uso dos jogos não garante um entendimento pleno do conteúdo de Orientação a Objetos, com a literatura dando ênfase no auxílio que o seu uso proporciona à compreensão da matéria.

3. Descrição do jogo na atividade

O jogo *Manuel Adventures* é composto de 4 (quatro) áreas, cada qual abrangendo um tópico principal da Disciplina Orientação a Objetos.

- 1ª fase: Abstração
- 2ª fase: Encapsulamento
- 3ª fase: Herança
- 4ª fase: Polimorfismo

Cada área é composta de mapas interligados subdividindo-as em etapas. Essas etapas possuem elementos de *gameplay* característicos que interagem com o jogador e permitem que o mesmo cumpra desafios ou receba recompensas. Na Figura 1 estão representados exemplos de seção com plataformas, onde o jogador deve passar por obstáculos para

¹<https://beieducacao.com.br/metodo-tradicional-de-ensino-e-metodologias-ativas-conheca-as-principais-diferencas/>

passar à etapa seguinte. Espalhados pela cena estão itens coletáveis, que o jogador pode juntar para somar à pontuação em *Manuel Adventures*. Além disso, os cenários de cada mapa são gerados a partir de *tilesets* específicos, e os mesmos permitem a identificação de qual dos tópicos, explicitados acima, os mapas pertencem, através de uma padronização conforme ilustrado na Figura 2. Há etapas especiais onde o jogador deve responder corretamente às perguntas para prosseguir, a mais comum está representada na Figura 3a onde o caminho correto será relevado após o aluno responder corretamente uma pergunta relacionada à OO. Responder corretamente leva a pontuações maiores, facilitando a jornada do personagem nas etapas seguintes. Nota-se que, de maneira a testar os conhecimentos dos alunos, esses questionários estão incorporados diretamente ao *level design* (figura 3). Outro exemplo dessa incorporação inclui o uso de plataformas onde o jogador deve se apoiar para responder a questão dada na cena, que estarão seguras de acordo com a resposta escolhida (Figura 3b).

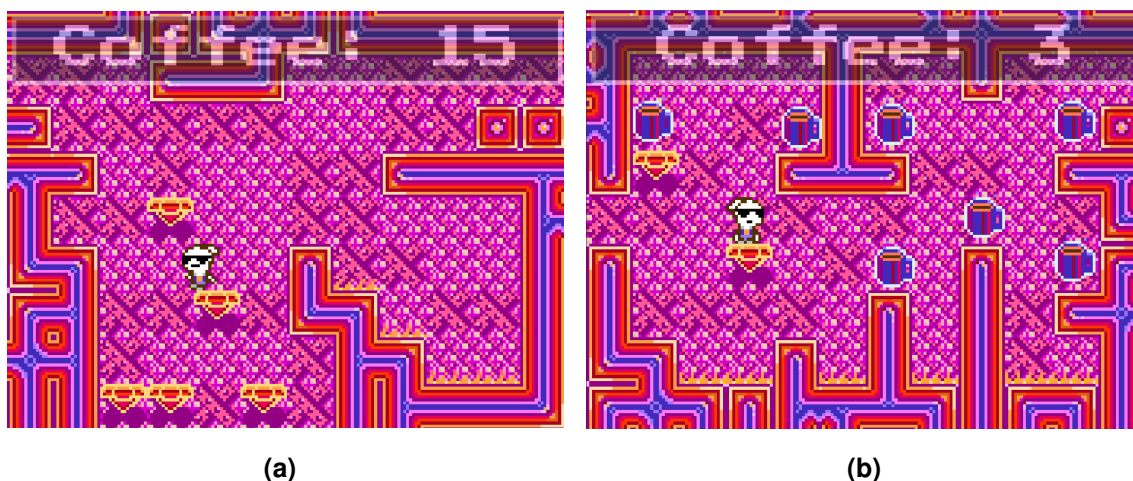


Figura 1. Ilustração dos mapas que compõem as fases. Em (a) um mapa seção de plataformas. Em (b) uma seção com coletáveis

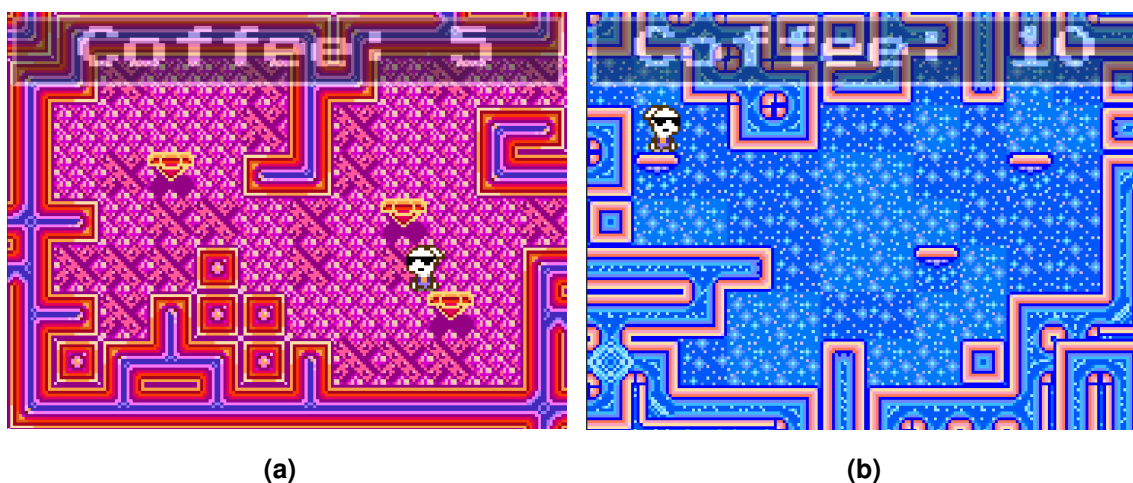


Figura 2. Ilustração dos mapas padronizados por *tilesets*. Em (a) um mapa predominantemente vermelho representando Abstração. Em (b) um mapa predominantemente azul e laranja, representando Encapsulamento

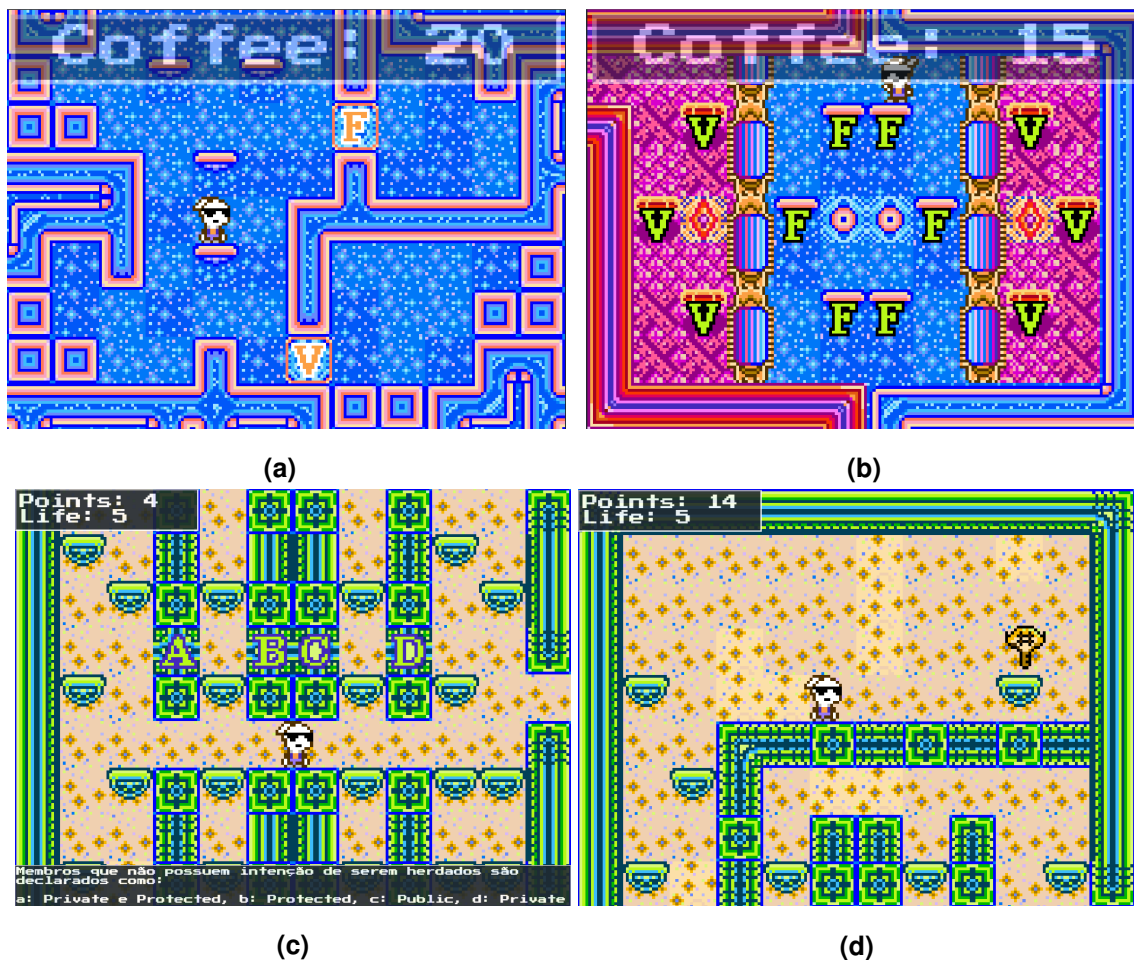


Figura 3. Ilustração dos questionários incorporados diretamente ao *level design*. Em (a) as respostas são indicadas por blocos bloqueando passagens. Em (b) as alternativas estão indicadas nas plataformas. Em (c) as alternativas levam a diferentes caminhos. Em (d) as chaves recompensam por responder corretamente uma pergunta

Cada área possui um cenário final como objetivo, e o mesmo é desbloqueado após a obtenção de determinada quantidade de chaves. Essas chaves podem ser obtidas da seguinte maneira: responder corretamente as questões da presente fase, ou seja, uma mecânica que torna obrigatório o teste de conhecimento do jogador. A quantidade a ser respondida de forma correta deve ser relativa a em média 60% do número total de questões. Porém, de forma que o jogo não se torne uma atividade maçante, caso o jogador deseje pular questões, ele poderá uma vez por fase. Ao obter todos os coletáveis é possível trocá-los por uma chave do nível atual, assim o aluno terá como custo para pular questões a exploração minuciosa da fase. Ademais, os coletáveis podem ser adicionados como bônus à pontuação ao final de uma fase caso não sejam utilizados para pular questões, recompensando jogadores dedicados tanto em conhecimento quanto em exploração.

Como mencionado anteriormente, existe um mapa final objetivo por fase, esse mapa contém um inimigo chefe, representante do tópico principal da fase. Em um combate com o chefe, são testados os conhecimentos adquiridos pelo jogador, com 5 a 10 questões a serem respondidas. O inimigo ataca em padrões pré definidos, porém a área de dano varia de acordo com as respostas para as questões atuais. O jogador estará seguro

em plataformas com as respostas corretas por exemplo (Figura 3b). Existem 2 formas de diminuir a vida do inimigo chefe, atacando-o diretamente sem tentar responder as perguntas, ou sobrevivendo às perguntas e atacando quando estiver vulnerável. O custo por atacar diretamente é um dano menor causado a ele e um risco maior do personagem perder vida. Ou seja, o jogador que responder corretamente será bonificado, não irá tomar dano e deixará o oponente vulnerável.

O projeto do jogo foi desenvolvido sem o auxílio algum de *game engines*, programas que auxiliam o desenvolvimento de jogos eletrônicos e tipicamente possuem motores gráficos e de física pré-implementados. Portanto, ao optar por não utilizar uma, gerou-se uma maior liberdade por parte dos desenvolvedores em realizar a implementação própria das partes gráficas e simulações de física. O ambiente de trabalho que permitiu tal liberdade foi o editor de texto gratuito e extensível Visual Studio Code ¹. Dentre as possibilidades de linguagens estudadas como viáveis para o projeto, escolheu-se HTML, CSS e Typescript, primeiramente pela sua facilidade de *deploy*, tornando o jogo mais acessível, com a necessidade somente de um *browser* para seu acesso. Além disso pelo vasto ecossistema do Typescript, e pela familiaridade dos integrantes com as linguagens, optou-se como escolha definitiva pelas mesmas. Por fim, os *sprites* foram criados através da ferramenta GraphicsGale ² devido a simplicidade da mesma, e seus *features* que auxiliam na criação de animações. E o controle de versões foi feito através do prático e popular Git ³.

4. Trabalhos Relacionados

Nesta seção são apresentadas outras literaturas que demonstram o auxílio de jogos e gamificação no ensino. Todos os trabalhos a seguir buscam motivar os alunos de algum modo em suas respectivas áreas de atuação. Alguns desses utilizam jogos de forma séria, na qual a abordagem primária não foca no entretenimento, sendo um meio de ensino [Michael and Chen 2005]. Propostas desse tipo são as que mais se aproximam do presente artigo.

Na categoria de jogos sérios o trabalho de [Queiros et al. 2022], propõe a exploração de conceitos geométricos por professores e alunos através de recursos educacionais digitais. A proposta se dá em vista de que não há modelos tridimensionais adequados e variados suficientemente para um ensino de qualidade que permita visualizar conceitos geométricos e incentivar participação crítica. Disponível como um jogo digital para dispositivos móveis, o projeto é concebido com foco no ensino e aprendizagem de projeções bidimensionais e tridimensionais, e visa fornecer aos estudantes uma forma de exercitar sua percepção visual de conceitos abstratos. Em sua jogabilidade os alunos devem concluir labirintos enquanto exercitam a visualização de formas geométricas. Os resultados apresentados indicam conformidade com a legislação educacional e boa avaliação do material pedagógico por professores. Porém, na avaliação estudantil nota-se média baixa em quesitos relativos a engajamento com o jogo, sendo considerado "muito parado". Logo a proposta apresentada cumpre os objetivos de ensinar mas não consegue manter um engajamento total do aluno.

POOkémon [Mombach et al. 2018], criado por alunos do Instituto Federal de

¹<https://code.visualstudio.com/>

²<https://graphicsgale.com/us/>

³<https://git-scm.com/>

Educação, Ciência e Tecnologia de Brasília, é um jogo educativo criado para auxiliar no ensino de orientação a objetos. Inspirado pelos antigos jogos de Pokémon, o jogo substitui as quatro habilidades dos personagens durante as batalhas por respostas de questões referentes à disciplina. O jogo ainda possibilita o professor alterar as perguntas como desejar.

Odyssey of Phoenix (OoP) [Wong et al. 2018] é um jogo *mobile* de RPG bidimensional desenvolvido com a proposta de conduzir os alunos a um aprendizado de programação orientada a objetos de forma simples, divertida e interativa. O design de OoP exige que os jogadores resolvam o jogo etapa por etapa de modo que possam adquirir o conhecimento sobre o paradigma de POO ao final de cada tarefa. Os autores levaram em conta várias teorias importantes como Teoria da Motivação da Atenção, Relevância, Confiança e Satisfação (ARCS), Aprendizagem Cognitiva, Prática Distribuída, Memória Episódica, para assegurar que o processo de aprendizagem no jogo seja simples, divertido, iterativo e motivador. O diferencial de OoP para outros jogos sérios voltados ao ensino dos conceitos de POO como CodeCombat, Alice 2D e Greenfoot, que requerem que o estudante tenha conhecimento sobre linguagens de programação como Java, Javascript e Lua; é que o aluno não precisa ter conhecimento prévio de nenhuma linguagem de programação.

OOP Serious Game [Lotfi and Bouhorma 2018] é um jogo sério que usa quebra cabeças, uma *gameplay* de *drag and drop* e um ambiente de zoológico, onde o jogador cria e classifica animais; para ensinar os conceitos básicos de programação orientada a objetos. O jogo é composto por quatro níveis, em que cada um o jogador deve completar uma série de tarefas para entender um dos quatro conceitos principais de OO, podendo o jogador ganhar prêmios ou ser penalizado de acordo com seu desempenho. O desempenho é medido de acordo com o tempo gasto para completar cada tarefa e pela quantidade de tarefas completas de forma correta.

Ainda na linha de jogos sérios, [Thaís Ferreira 2021] propõe ampliar o conhecimento do recente tema Ecossistemas de Software (ECOS) contido na área de conhecimento Engenharia de Software (ES). Por sua temporalidade, o assunto não possui referenciais bibliográficos didáticos suficientes, além de ser um conceito considerado extremamente abstrato, tornando seu ensino desafiador. Técnicas de *storytelling*, contar, desenvolver e adaptar histórias foram aplicadas a um jogo do gênero plataforma interligando os conhecimentos a ensinar com fases e missões contidas na narrativa. Os resultados indicaram avaliação muito boa nos quesitos diversão, satisfação e relevância principalmente em grupos compostos de usuários com nenhuma ou moderada experiência prévia no tema ECOS. Porém assim como na proposta de [Queiros et al. 2022], imersão e jogabilidade pecam, mecânicas mais complexas e tarefas variadas para uma maior imersão permanecem como um desejo.

De forma a explorar a performance do aprendizado dos alunos, o trabalho de [Oren et al. 2021] apresenta três ferramentas *in-game* para auxiliar na curva de aprendizado e na experiência de *gameplay*. Em um contexto geral, foram testadas as seguintes técnicas: um guia de instruções específico ao tema, sistemas de apoios temporários e funções para aumentar a produtividade, eliminando tarefas repetitivas. Testou-se os conceitos aplicados diretamente ao jogo educacional de engenharia "Planet K", que objetiva ensinar conceitos de circuitos digitais. Verificou-se que o *design* dos recursos do jogo

influenciam diretamente no desempenho dos estudantes, portanto, uma análise cuidadosa é necessária a fim de avaliar se o uso de uma certa ferramenta trás impactos favoráveis ao aprendizado, e é suscetível à adaptações para atender certas demandas. Os resultados do estudo apontam que o uso de guias e sistemas de apoio na resolução de problemas tem maior influência na performance dos estudantes, de tal modo que grande parte dos obstáculos impostos são ultrapassados.

Em um viés alternativo à presente proposta, o trabalho de [João Stephan 2020] adota uma abordagem de desenvolvimento de jogos, sem requerer dos alunos o uso de ferramentas específicas, utilizando somente a linguagem de programação C estudada no âmbito da disciplina, e o auxílio da biblioteca gráfica SFML (Simple and Fast Multimedia Library). Em virtude de alguns alunos não possuírem conhecimento prévio em programação ou pensamento lógico, algo que pode acarretar em problemas no aprendizado inicial, propõe-se avaliar essa abordagem. A mesma, dita como uma tática de apoio ao ensino e aprendizagem de programação em disciplinas introdutórias dos cursos de Ciência da computação e áreas afins. Os resultados obtidos pela experiência são em geral positivos, principalmente no âmbito motivação, e a maior parte das dificuldade dos alunos demonstrou-se mesmo pela falta de conhecimento prévio. Em contrapartida neste trabalho o tamanho amostral foi limitado, algo não ideal estatisticamente. De toda forma, a obra demonstra que elementos lúdicos atrelados a formas de ensino podem prender a atenção de estudantes e motivar a buscar mais sobre os temas ensinados.

Na mesma linha de abordagens com desenvolvimento temos o jogo Pond, um jogo da série Block Games do [Google 2022], tem como objetivo ensinar programação de uma forma divertida, com foco principalmente em crianças que não tem experiência prévia na área. O jogo consiste em uma lagoa com patos, onde o jogador deve programar o pato mais esperto. Os patos disputam entre si utilizando um canhão para atingir os outros, até que a energia dos adversários chegue a zero e ele seja o último ainda nadando. O jogador pode programar seu pato através de um *script* em JavaScript ou Blockly. Esse *script* é escrito antes da partida começar e, após o início da partida, não pode ser alterado e serve como a inteligência artificial do pato do jogador. O jogo pode ser jogado *online* ou pode ser baixado para jogar localmente. Todo o código é *open-source*, podendo ser modificado livremente para servir a um propósito específico.

Assim como os primeiros trabalhos apresentados, a concepção do presente artigo se dá como um jogo sério, utilizando o produto como um meio de entregar conhecimentos, porém na contramão de [Queiros et al. 2022] e [Thaís Ferreira 2021], o presente projeto visa uma interação mais dinâmica com o jogador. Pontos negativos apresentados nas propostas citadas incluem falta de dinâmica, pouca variedade em mecânicas, e baixa complexidade na interseção entre conhecimentos e jogabilidade. Esses fatores se mostraram mais flexíveis em atividade de desenvolvimento direto, pois a mesma permite a inovação do programador. Logo é crucial que o mesmo deva ser capaz de exercitar sua liberdade criativa ao mesmo tempo que absorve conhecimentos e se mantém entretido. Portanto, aqui, assim como descrito na seção 2, propõe-se uma metodologia de jogos sérios mais ativa e complexa, incorporando os conhecimentos e perguntas relativas ao assunto estudado diretamente no *level design* e outros elementos da jogatina.

5. Análise Comparativa

Na mesma linha do projeto desenvolvido, é comparado este projeto com os trabalhos apresentados na seção anterior. A análise é feita sobre o objetivo final do projeto, as estratégias empregadas para melhorar o engajamento, metodologia de ensino, contexto de experimentação, público-alvo e *frameworks* utilizados.

A respeito do objetivo a ser alcançado, a maioria dos trabalhos visam auxiliar o aprendizado do aluno através de uma maneira descontraída e moderna usando um jogo virtual. É importante ressaltar que nenhum dos trabalhos analisados pretende substituir inteiramente o ensino básico, apenas reforçar o conteúdo aprendido em sala de aula. Outros trabalhos visam estudar as ferramentas utilizadas para facilitar o aprendizado dos jogadores, evidenciando que o método empregado no ensino é importante para cumprir o objetivo proposto. O público-alvo também é um parâmetro guiado pelo objetivo do projeto, para alguns trabalhos o grau de conhecimento do aluno é relevante para definir o conteúdo programático, os obstáculos e a metodologia de ensino, conforme evidencia [Wong et al. 2018].

Nota-se um tema recorrente nos trabalhos analisados a respeito da motivação dos alunos. A conciliação entre diversão e aprendizado é um desafio comum no tema de jogos sérios, com, por exemplo, o trabalho de [Queiros et al. 2022] que possui uma curva de aprendizado positiva, porém o quesito engajamento peca em comparação. Outros projetos estudados têm essa característica de engajamento bem avaliada, os alunos possuem interesse no assunto passado ou estão entretidos com o jogo, porém possuem um ensino muitas vezes confuso ou raso. Alternativas são desenvolvidas para contornar o problema, seja trabalhando no *design* de jogo ou na criação de ferramentas auxiliares ao ensino, com eficiência mostrada em diversos trabalhos estudados, sobretudo sobre pessoas com menor dominância no assunto. Percebe-se que não há uma concordância à respeito do método utilizado, evidenciando que jogos sérios ainda estão nas etapas iniciais de seu desenvolvimento.

A respeito da metodologia de ensino empregada, os trabalhos possuem a característica comum de ensinar através de recursos audiovisuais. Utilizar obstáculos como forma de guiar ou instruir o aluno durante a jogatina é recorrente a todos os projetos analisados. Apesar desse aspecto compartilhado, os jogos sérios trazem inovações na metodologia de aplicação deles, como é o caso do [Mombach et al. 2018], onde define-se o problema como um inimigo que deve ser superado através dos conhecimentos do jogador. Mesmo que não tenham resultados conclusivos a respeito de como essa mecânica deve ser aplicada de maneira a levar a uma maior curva de aprendizado do aluno, bem como qual procedimento deve ser aplicado de forma a mantê-lo engajado no projeto, nota-se que jogos mais interativos, dinâmicos e fluidos possuem uma recepção mais positiva do público-alvo.

O contexto de experimentação foi definida envolvendo o público-alvo e o objetivo final para grande parte dos trabalhos relacionados. Em alguns trabalhos, analisa-se uma experimentação ao longo do desenvolvimento do projeto, seguido por uma avaliação dos usuários durante a implementação. Essa participação em alguns projetos é voluntária, dificultando a coleta de dados significativos para comparação e análise precisa [João Stephan 2020]. Em outros projetos, a experimentação envolve

Para a maioria dos projetos, o *feedback* coleta dados acerca da motivação dos alunos, da compreensão e da satisfação. Nesses casos, o retorno geralmente foi positivo, principalmente a respeito da satisfação, ao passo que o entendimento ficou a desejar. Apesar disso, nota-se uma análise mais cuidadosa dos resultados em [Wong et al. 2018], com uma avaliação sendo feita antes e após o período de testes sob um grupo cuidadosamente selecionado, no caso, estudantes com notas baixas.

Na tabela 1 são apresentadas resumidamente as características analisadas para cada um dos trabalhos relacionados e com o projeto desenvolvido neste artigo. Na primeira coluna temos os nomes dos projetos percorridos na seção 4. Para cada um dos trabalhos, colocou-se um breve resumo do objetivo do projeto, estratégia utilizada, metodologia, experimentação, público-alvo e *frameworks*.

| Projeto | Objetivo Final | Estratégia de Engajamento | Metodologia | Experimentação | Público-alvo | Framework |
|--------------------------|---|----------------------------|---|---|--|-----------------------------|
| Manuel Adventures | Auxiliar no ensino de OO | Uso de jogo <i>desktop</i> | Responder às perguntas | Exposição do protótipo | Alunos de OO | Não utiliza |
| Enigmas de Vocabulário | Auxiliar o ensino de Geometria | Uso de jogo <i>mobile</i> | Identificar figuras geométricas | Avaliações envolvendo participantes | Alunos do Ensino Médio e educadores | Não mencionado |
| POOkemón | Auxiliar no ensino de OO | Uso de jogo | Responder às perguntas | Questionário de avaliação | Alunos de OO | JavaFX e Glue Scene Builder |
| Odyssey of Phoenix (OoP) | Auxiliar no ensino de OO | Uso de jogo <i>mobile</i> | Completar missões temáticas | Avaliação sobre OO realizados antes e depois de jogar | Alunos de OO selecionados | Game-based learning |
| OOP Serious Game | Ensinar OO | Uso de jogo | Solucionar <i>puzzles</i> | Análise sobre resultados <i>in-game</i> | Desenvolvedores de <i>software</i> novatos | Unity |
| Arvore de ECOS | Apoiar o ensino de ECOS | Uso de jogo | Uso de metáforas e elementos narrativos | Avaliações envolvendo participantes | Alunos de Engenharia de Software | Construct 2 |
| Planet K | Analisar o uso de ferramentas no ensino | Uso de jogo | Solucionar problemas com ferramentas <i>in-game</i> | Análise sobre resultados <i>in-game</i> | Alunos de graduação | Não mencionado |
| GameProgJF | Apoiar o ensino de programação | Criação de jogo | Implementação de mecânicas ao jogo | Preenchimento opcional de formulário | Alunos de graduação | Biblioteca SFML |
| Pond | Ensinar programação básica | Uso de jogo | Implementação de mecânicas ao jogo | <i>Feedback</i> opcional via e-mail | Iniciantes em programação | Não mencionado |

Tabela 1. Características Comparativas entres os Trabalhos Relacionados

6. Considerações Finais

Este artigo apresentou o Manuel Adventures, que tem como objetivo auxiliar no ensino de Orientação a Objetos. O projeto foi testado somente pela equipe interna, e sugestões foram levantadas pelos membros dessa. As mesmas incluem maior variedade de inimigos, de perguntas e planejar um uso mais adequado aos coletáveis. Para resultados mais concretos uma versão do mesmo deverá ser aplicada e avaliada por alunos. Logo, planeja-se para um futuro próximo o lançamento do jogo ao público aberto para que receba *feedback* da comunidade externa.

Como trabalho futuro, o jogo continuará sendo desenvolvido, a nova versão contará com uma maior quantidade de questões e mecânicas de jogabilidade mais diversificadas. Já em andamento, consiste a criação de novas áreas, subdivisões dos 4 pilares da Programação Orientada a Objetos, a fim de que assuntos específicos possam ser tratados de uma melhor forma. Além disso, para verificar a eficácia do ensino, pretende-se um estudo experimental, no qual alunos da disciplina possam jogar e serem avaliados quanto a seu aprendizado. Esse estudo será feito com o auxílio de um professor responsável por ministrar Orientação a Objetos no semestre que for aplicado, e permitirá conclusões mais realistas sobre o apoio do ensino gamificado.

Referências

- Caillois, R. and Barash, M. (1961). *Man, Play, and Games*. Free Press of Glencoe.
- Cleitom José Richter, G. B. (2018). Apoiando o ensino de orientação a objetos por meio do desenvolvimento de jogos. *Tecnologias na Educação*, 28.
- de Freitas, S. (2018). Are games effective learning tools? a review of educational games. *Journal of Educational Technology Society*, 21(2):74–84.
- Eliane M. Duarte de Barros, E. C. F. (2013). O paradigma do desenvolvimento orientado a objetos a sua dificuldade de aprendizado. *Proficiência*, 3.

- Google (2022). Blockly games: Pond. <https://blockly.games/pond-duck>. Acessado: 2022-06-09.
- João Stephan, Alessandra Oliveira, M. R. (2020). O uso de jogos para apoiar o ensino e aprendizagem de programação. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 381–390, Porto Alegre, RS, Brasil. SBC.
- Lotfi, E. and Bouhorma, M. (2018). Teaching object oriented programming concepts through a mobile serious game. pages 1–6.
- Mattar, J. and Nesteriuk, S. (2016). Estratégias do design de games que podem ser incorporadas à educação a distância. *RIED-Revista Iberoamericana de Educación a Distancia*, 19(2):91–106.
- Michael, D. R. and Chen, S. L. (2005). *Serious Games: Games That Educate, Train, and Inform*. Muska Lipman/Premier-Trade.
- Mombach, J., Castro, B., Santos, M., and Santos, E. (2018). Pookémon: um jogo sobre programação orientada a objetos.
- Oren, M., Pedersen, S., and Butler-Purry, K. L. (2021). Teaching digital circuit design with a 3-d video game: The impact of using in-game tools on students’ performance. *IEEE Transactions on Education*, 64(1):24–31.
- Ortiz, J. d. S. B. and Pereira, R. (2021). Computational thinking for youth and adults education: model, principles, activities and lessons learned. *Revista Brasileira de Informática na Educação*, 29:1312–1336.
- Queiros, L. M., Gomes, A. S., Pereira, J. W., Castro Filho, J. A. d., Santos, E. M. d., and Silva Neto, D. F. d. (2022). Enigmas de yucatàn: Recurso educacional digital para o ensino de geometria espacial. *Revista Brasileira de Informática na Educação*, 30:108–134.
- Rafael Lucas da Costa, Carlos Eduardo Ribeiro, D. d. F. G. T. J. R. M. (2021). Desenvolvimento de jogos como estratégia para aprendizagem de programação orientada a objetos. *Ensino, Pesquisa e Desenvolvimento na Engenharia Eletrônica e Computação*, pages 49–60.
- Thaís Ferreira, Davi Viana, R. S. (2021). Árvore de ecos: Um jogo para ensino de conceitos de ecossistemas de software. *Revista Brasileira de Informática na Educação*, 29(0):273–300.
- Wong, Y. S., Yatim, M., and Tan, W. H. (2018). Learning object-oriented programming paradigm via game-based learning game – pilot study. *The International journal of Multimedia Its Applications*, 10:181–197.