

CODE_DUNGEON: UM SERIOUS GAME PARA AUXILIAR NO APRENDIZADO DE PROGRAMAÇÃO

Gabriel C. Oliveira¹, Elisa Boff¹

¹Área de Ciências Exatas e Engenharias – Universidade de Caxias do Sul (UCS)
Caxias do Sul – RS – Brasil

{gcoliveira7, eboff}@ucs.br

Abstract. *The dropout rate in undergraduate courses in the area of technology is quite high when compared to other areas of knowledge. One of the factors that contribute to this high rate is the difficulty students have with learning programming, especially in introductory subjects. These difficulties normally appear in the basic concepts of the discipline, such as abstraction and logic, for understanding algorithms, and functions and repetition structures, for understanding programming itself. These problems could be mitigated by teaching computational thinking in basic education classes. Proposals for curricula for this purpose, such as the CIEB, exist but they are not yet applied. In view of the difficulties of undergraduate students with learning programming, this work aims to propose and evaluate the creation of a serious game that can help in the development of Computational Thinking and, consequently, help the learning of the initial subjects of technology courses. The evaluations, carried out using the MEEGA method, were applied in computer programming classes to validate the application of the game as a tool to help teach programming and conditional structures.*

Keywords— *Serious Games, programming learning, computational thinking*

Resumo. *A taxa de evasão em cursos de graduação na área de tecnologias é bastante alta quando comparada a outras áreas do conhecimento. Um dos fatores que contribuem para essa taxa elevada é a dificuldade dos alunos com o aprendizado de programação, principalmente em disciplinas introdutórias. Essas dificuldades normalmente se apresentam nos conceitos básicos da disciplina, como abstração e lógica, para entendimento de algoritmos, e funções e estruturas de repetição, para entendimento da programação em si. Esses problemas poderiam ser amenizados com o ensino de pensamento computacional em turmas de educação básica. Propostas de currículos para essa finalidade, como o do Centro de Inovação para a Educação Brasileira (CIEB), existem, porém, eles ainda não são aplicados. Tendo em vista as dificuldades dos alunos de graduação com o aprendizado de programação, esse trabalho visa propor e avaliar a criação de um jogo sério que possa auxiliar no desenvolvimento do Pensamento Computacional e, conseqüentemente, auxiliar a aprendizagem das disciplinas iniciais de cursos de tecnologia. As avaliações, realizadas utilizando o método MEEGA+ (Model for the Evaluation of Educational Games), foram aplicadas em turmas de programação de computadores para validar a aplicação do jogo como ferramenta de auxílio no ensino de programação e estruturas condicionais.*

Palavras-chave— *Jogos sérios, aprendizagem de programação, pensamento computacional*

1. Introdução

Com a crescente transformação digital nas diversas áreas do conhecimento, a busca por profissionais da área de TI vem aumentando nos últimos anos. A procura por estes profissionais teve um crescimento de 671% apenas em 2020 [1]. Em resposta a isso, as instituições de ensino superior têm uma projeção de formar cinco vezes mais profissionais na área até 2025, em comparação aos dados de profissionais formados até 2019 [2]. Além disso, de acordo com o Semesp (Sindicato das Entidades Mantenedoras de Estabelecimentos de Ensino Superior no Estado de São Paulo), o curso de Sistemas de Informação está entre os 10 mais procurados na rede privada e entre os 20 mais procurados da rede pública [3]. Outra estatística, porém, vai de encontro a essa. Este curso possui a maior taxa de evasão, na modalidade presencial, e está entre os 10 primeiros nessa mesma estatística para modalidades a distância [4].

Um dos grandes desafios para alunos que estão ingressando em cursos da área é o aprendizado de programação. Disciplinas introdutórias de programação normalmente encontram-se nos semestres iniciais dos cursos da área e acabam se tornando uma barreira para alguns alunos, por diversos fatores. Possíveis motivos para explicar esse problema variam desde a dificuldade para compreender conceitos de programação, até a falta de motivação para o aprendizado da mesma. Possíveis soluções também são bastante variadas, envolvendo projetos de ambientes de aprendizagem e jogos sérios para auxiliar na educação de programação.

A falta de conhecimento prévio sobre programação e algoritmos também pode ser uma barreira para alunos de graduação. Por isso este trabalho buscou entender, por meio do currículo desenvolvido pelo CIEB, se pode ser aplicado o ensino de pensamento computacional para turmas de ensino básico.

Tendo como proposta principal um jogo sério para ensinar e desenvolver conceitos de programação, o trabalho também buscou modelos para desenvolvimento e avaliação de jogos educativos como ferramentas de ensino, a fim de fundamentar o jogo proposto. Essas ferramentas, como o SGDA [6] e o MEEGA+ [7], foram utilizadas como método de avaliação e validação da proposta, buscando entender os fundamentos de um jogo sério e como ele pode ser utilizado para o ensino.

2. Pensamento Computacional e Jogos Sérios

No cenário ideal para o ensino de programação em disciplinas iniciais de cursos de graduação, os alunos deveriam ingressar tendo uma noção básica sobre pensamento computacional, que envolve ensino de abstração, algoritmos, decomposição de algoritmos e reconhecimento de padrões. Para avaliar esse cenário “ideal” do ensino de programação, este trabalho investigou currículos de pensamento computacional para turmas de ensino básico e médio, conforme proposto pelo CIEB (Centro de Inovação para a Educação Brasileira).

O Currículo de Referência em Tecnologia e Computação, desenvolvido pelo CIEB, propõe currículos para ensino infantil e médio, contemplando temas de tecnologia e computação. Ambos os currículos propostos possuem eixos que buscam o ensino de Pensamento Computacional. No currículo de referência para educação infantil e ensino fundamental, a proposta do eixo de Pensamento Computacional explora os conceitos de Abstração, Algoritmos, Decomposição e Reconhecimento de Padrões. Para a proposta curricular de Ensino Médio, o eixo de Pensamento Computacional possui oito Unidades

Curriculares (UCs), sendo quatro essenciais, totalizando 280 horas, e quatro eletivas, totalizando 240 horas. As UCs essenciais envolvem Programação de Computadores, Tecnologias para Internet, Design de Aplicativos e Ciência de Dados. Já as eletivas envolvem Jogos Digitais e Analógicos, Internet das Coisas, Simulações de Fenômenos Naturais e Robótica.

Souza *et al.* [8] apresentam uma revisão sistemática da literatura com vistas a analisar os problemas com o ensino de programação. Os autores apontam como possíveis soluções para resolver esta dificuldade de aprendizagem, a utilização de técnicas de visualização, onde busca-se apresentar o ambiente de programação em um formato que seja mais divertido e não ameaçador. Além de *serious games* e ambientes de programação pedagógicos, que são ferramentas que possibilitam a criação e execução de programas, mas com um maior foco no ensino e na aprendizagem.

Apesar de ser um livro com foco em jogos sérios de carta e tabuleiros, as definições apresentadas em “Serious Games” [5] ainda se encaixam em jogos sérios digitais. Abt [5] define que “O paradoxismo de serious games une a seriedade do pensamento e dos problemas que o exigem com a liberdade experimental e emocional do jogo ativo. Serious games combinam a concentração analítica e questionadora do ponto de vista científico com a liberdade intuitiva e as recompensas de atos artísticos imaginativos”.

4. Metodologia

Para o desenvolvimento do jogo CODE_DUNGEON, foi usada a ferramenta Unity3D, tendo como base para o design do jogo os elementos propostos pelo framework SGDA. Para a avaliação do jogo desenvolvido, foram feitos *playtests* do jogo com o público-alvo; utilizando o questionário desenvolvido pelo método MEEGA+ para poder quantificar a satisfação dos jogadores que participaram da sessão de testes do jogo. Além do MEEGA+, foi desenvolvido um segundo questionário para os jogadores, que tem como objetivo avaliar o aprendizado dos conteúdos desenvolvidos pelo jogo.

4.1. O framework SGDA

Tendo em vista a falta de ferramentas para avaliação de design focadas em jogos sérios, Mitgutsch e Alvarado [6] desenvolveram um framework para avaliação de design para jogos sérios, o SGDA (Serious Game Design Assessment). O framework SGDA propõe uma ferramenta para que os conceitos de game design possam ser mais bem estudados e avaliados quando aplicados a um jogo sério. O framework é composto por seis principais componentes para a estrutura conceitual de um jogo sério:

- a) **Propósito:** a primeira etapa que deve ser levada em conta na hora de projetar um jogo sério é o propósito dele, identificando qual ideia ou mensagem o jogo deve passar para o jogador.
- b) **Conteúdo:** o conteúdo e informação de um jogo sério são os dados, estatísticas e outras informações pertinentes que são apresentadas ao jogador. Além disso, esses dados podem ser separados e examinados de acordo com o contexto em que se apresentam
- c) **Mecânica:** as mecânicas são o principal meio pelo qual o jogador vai interagir com o jogo, elas englobam o objetivo central do jogo, como funciona o sistema de recompensas, quais os principais obstáculos/desafios, como funciona o balanceamento de dificuldade e quais as condições de vitória

d) **Ficção e Narrativa:** a narrativa e o ambiente de um jogo sério devem levar em conta o propósito do jogo, buscando elevar a mensagem ou ideia que o jogo quer levar aos jogadores.

e) **Estética e Gráficos:** a estética de um jogo sério não é o primeiro elemento a ser projetado, porém é o primeiro elemento que vai ser apresentado ao jogador, portanto “desempenha um papel fundamental na introdução do propósito do jogo e seu impacto no jogador”. Esse elemento não leva em conta apenas os gráficos do jogo, mas também toda a linguagem audiovisual, desde o som, imagem e a presença ou não de animações *in-game* que apresentam o contexto ficcional.

f) **Enquadramento:** além dos itens anteriores, se faz necessário pensar na maneira em que esses elementos são apresentados, considerando o público-alvo. Para isso, o framework levanta questões como “O público-alvo tem problemas com controles, interface ou com a ficção do jogo? Quais são as habilidades necessárias? [...] Os níveis de dificuldade estão balanceados de acordo com as necessidades do público?” Um jogo sério deve ser de fácil acesso ao público, mas ainda assim precisa engajar o jogador com um gameplay balanceado.

O objetivo do Framework SGDA não é somente apresentar e analisar os elementos fundamentais para um jogo sério, mas também mostrar como esses sistemas interagem com o jogo e entre si para levar a melhor experiência para o jogador. Circulando ao redor do propósito do jogo, todos esses elementos devem coexistir de forma a introduzir ao jogador as ideias e conceitos que o jogo sério pretende abordar. De acordo com [6], “Se os jogos sérios são projetados para serem projetos baseados em propósitos e pretendem impactar seus jogadores, seu propósito precisa ser considerado em todos os componentes de design.”

4.2. O modelo MEEGA+

O modelo de avaliação MEEGA+ se concentra em jogos educacionais. Na proposta de Petri, Wangenheim e Borgatto [7], afirma-se que jogos educacionais são uma estratégia bastante válida cientificamente, porém existem poucas ferramentas que possibilitem a avaliação desses jogos. Uma dessas possíveis avaliações é o modelo MEEGA (Model for the Evaluation of Educational GAMES), proposta em 2011. O modelo MEEGA+ é uma proposta de evolução com base na proposta original.

Tendo como referência oito trabalhos que avaliam design de jogos sérios educacionais, o MEEGA+ considera sua própria base de qualificação. São analisados framework, escalas e metodologia e identificadas possíveis melhorias. Levando em conta a análise feita, o modelo MEEGA+ apresenta um questionário organizado em 9 dimensões, a saber: Usabilidade (Estética, Aprendizabilidade, Operabilidade, Acessibilidade, Proteção contra erros do usuário), Confiança, Desafio, Satisfação, Interação social, Diversão, Atenção focada, Relevância e Aprendizagem percebida. As respostas às questões são apresentadas em uma escala Likert de 5 pontos que varia de “discordo totalmente” a “concordo totalmente”, com objetivo de permitir a expressão de opinião com uma maior precisão.

5. Desenvolvimento

Nesta seção são detalhadas as etapas de desenvolvimento do jogo CODE_DUNGEON, de acordo com o framework SGDA, cujas etapas são: propósito, conteúdo, mecânicas, ficção e narrativa, estética e gráficos e enquadramento.

5.1. Propósito

O propósito principal do CODE_DUNGEON gira em torno do aprendizado de programação. O jogo desenvolvido busca levar o jogador a aprender conceitos de programação, como estruturas condicionais. Para isso, o game design foi feito de forma a ensinar esses conceitos ao jogador, mas também deixá-lo solucionar alguns dos problemas apresentados de forma livre, balanceando a quantidade de níveis que vão servir de tutorial para as mecânicas com níveis que ficarão abertos para o jogador aplicar os conhecimentos desenvolvidos até aquele ponto.

5.2. Conteúdo

O conteúdo é toda a informação apresentada ao jogador por meio da interface do jogo. No nível mais básico, é apresentada a classe de cada personagem, seus pontos de vida e quais habilidades esse personagem possui. Além disso, os métodos utilizáveis para o jogador desenvolver a programação de cada personagem também estão disponíveis na interface de jogo. Quanto às estatísticas presentes, o jogo poderá apresentar a quantidade de turnos que o jogador demorou para concluir cada nível, além de quantos métodos foram usados, tentando levar ao jogador a ideia de otimização do programa desenvolvido.

5.3. Mecânica

O objetivo principal de cada nível do CODE_DUNGEON é derrotar todos os inimigos presentes. O jogador tem seu grupo de personagens pré-determinados para cada nível do jogo e cada um desses personagens possui um conjunto de possíveis ações que podem ser tomadas. A interface do jogo apresenta um espaço para o jogador criar suas instruções de forma algorítmica, indicando qual a condição (como, por exemplo, quantidade de pontos de vida do inimigo) e qual a ação cada personagem deve tomar caso a condição seja verdadeira (atacar ou defender, por exemplo).

5.4. Ficção e Narrativa

A narrativa do jogo baseia-se em uma batalha com cenário medieval, procurando guiar o jogador pelas fases. Para isso, tutoriais e dicas ficam disponíveis para o jogador no início de cada fase, apresentando o objetivo ao jogador, demonstrando novas mecânicas e como cada função pode ser utilizada na programação dos personagens. Conforme o jogador avança pelas fases, o jogo disponibiliza uma quantidade maior de condições e de ações que podem ser escolhidas pelo jogador, além de simplificar as dicas no início de cada nível, de forma que o jogador tenha mais liberdade para descobrir a sua própria resposta para os obstáculos apresentados em cada nível.

5.5. Estética e Gráficos

Por se tratar de um jogo do gênero RPG, o jogo tem como tema principal da estética elementos medievais e de fantasia, se utilizando de pacotes de *assets* adquiridos para o desenvolvimento do jogo. Os elementos gráficos do jogo, sejam eles personagens, cenários ou fontes, usam a estética de *pixel art* (Figura 1).

Para os personagens, foram utilizados os pacotes de *assets* - elementos que constituem um jogo, como ilustrações e efeitos sonoros - MiniFolks, produzidos por LYASeeK¹. Os

¹ <https://lyaseek.itch.io/> Acessado em Julho de 2022.

cenários utilizam o pacote Tiny Tales: Overworld, desenvolvido por megatiles². As fontes utilizadas para as interfaces do jogo foram as fontes de pixel dos pacotes Inkbit e New Hi-Score, produzidos por Rodrigo BSL³.

5.6. Enquadramento

O público-alvo do jogo são pessoas com conhecimento iniciante em programação, ou que já possuem interesse no assunto. Portanto, o jogo assume que o jogador tenha um conhecimento básico sobre programação, mas ainda assim explica os conceitos a serem desenvolvidos de forma a ajudar o jogador no entendimento dos conteúdos. O balanceamento dos níveis do jogo é feito levando em consideração se o nível é um tutorial sobre uma nova mecânica ou um nível livre para o jogador desenvolver seu código. Em níveis de tutorial o jogo deve “segurar a mão” do jogador e ensinar como determinada mecânica funciona no ambiente do jogo. Já em níveis abertos ao jogador, os desafios são apresentados de forma mais aberta, com algumas dicas sobre como cada código pode ser usado em determinadas situações.

5.7. Implementação

O jogo foi desenvolvido utilizando a *engine* Unity3D, usando a linguagem de programação C# e os pacotes de *assets* e fontes apresentados na seção 5.5. O jogo possui cinco fases, sendo a primeira delas um tutorial para apresentar as mecânicas do jogo. Mostrando como funcionam as instruções programadas, e ensinando ao jogador como criar uma nova instrução para o personagem.

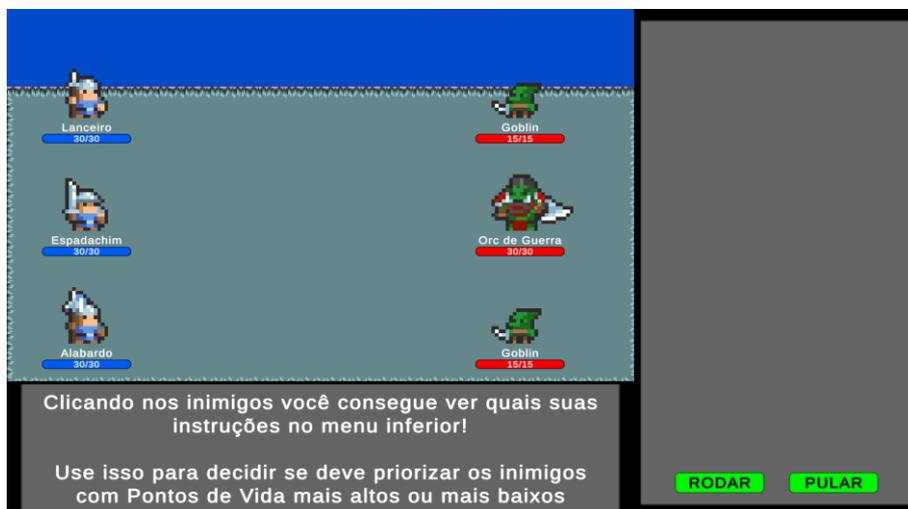


Figura 1. Início da segunda fase, com dicas sobre comportamento dos inimigos e possíveis instruções

A partir da segunda fase, o controle de criação de instruções para os personagens fica totalmente sob controle do jogador, porém o jogo ainda dá dicas no início de cada fase (Figura 1), além de introduzir novas mecânicas (Figura 2) e possíveis instruções para o

² <https://megatiles.itch.io/> Acessado em Julho de 2022.

³ <https://bsl.itch.io/> Acessado em Julho de 2022.

jogador (Figura 3). Durante a terceira e quarta fase, o jogador é apresentado a novas mecânicas, de cura e bloqueio, respectivamente. Para que possa utilizar essas mecânicas, o jogador tem a possibilidade de usar novas instruções, tendo seus aliados como alvo. A quinta fase serve como um teste para os conhecimentos desenvolvidos pelo jogador nas fases anteriores. Ela não apresenta nenhuma dica ao jogador, apenas o objetivo de derrotar todos os inimigos, permitindo ao jogador utilizar todas as mecânicas desenvolvidas nas fases anteriores.



Figura 2. Interface com as instruções do inimigo selecionado

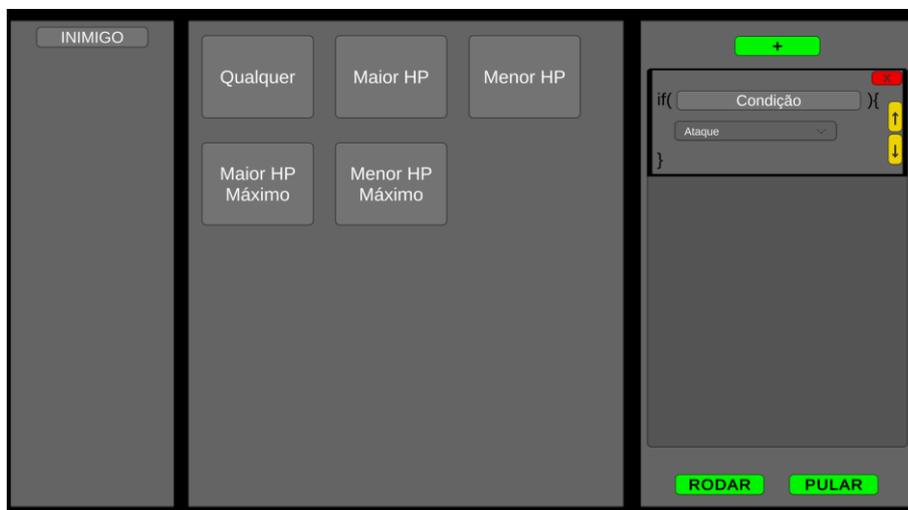


Figura 3. Criação de instruções na segunda fase do jogo

4. Validação e Resultados

Para a validação do jogo, foram feitos dois testes utilizando a ferramenta MEEGA+. No primeiro teste, com um grupo fechado de cinco pessoas, foi testada uma versão inicial do jogo, para identificar mudanças a serem feitas nas mecânicas e tutoriais, além de buscar um melhor balanceamento no sistema de combate. No segundo teste, aplicado com duas turmas de Programação de Computadores I, o jogo foi avaliado como ferramenta de ensino. Nesta seção, será descrita a aplicação dos questionários do MEEGA+, além de

apresentar os resultados dos testes realizados, utilizando algumas das respostas coletadas no teste de validação.

4.1. Pré-teste

A etapa de pré-teste foi aplicada utilizando uma versão ainda incompleta do jogo, que possuía apenas a fase de tutorial e uma fase “livre” para o jogador explorar as mecânicas. Essa etapa foi aplicada com um grupo de cinco pessoas experientes em jogos e programação, e tinha como objetivo identificar *bugs* e falhas no jogo, além de coletar um primeiro feedback sobre possíveis alterações nas mecânicas do jogo.

Além da correção dos bugs identificados na etapa de pré-teste, os principais *feedbacks* desse primeiro teste foram sobre a dificuldade de compreender totalmente as mecânicas do jogo que, por estar numa versão inicial, não eram introduzidas a cada fase, como é na versão final do jogo. Para auxiliar na compreensão do jogador, além de introduzir mecânicas e instruções novas a cada fase, também foram adicionadas as dicas ao início de cada fase para guiar o jogador à solução, mas ainda o desafiando a desenvolver a resposta. Além disso, após esta etapa de pré-teste foi alterado o balanceamento do combate (números dos pontos de vida e ataque dos personagens), as animações dos personagens foram otimizadas e foi adicionado um botão de “pular”, que é apresentado ao jogador no tutorial e acelera as ações de cada turno da batalha.

4.2. Validação utilizando o MEEGA+

Para a validação do jogo sério como potencial recurso de aprendizagem, o formulário MEEGA+ foi aplicado com duas turmas de Programação de Computadores I. No total, entre as duas turmas, 28 alunos participaram do processo de validação do jogo. Os testes foram realizados em trinta minutos e divididos em duas etapas. Na primeira etapa, durante os primeiros vinte minutos da aplicação, os alunos jogaram o jogo individualmente. Durante a etapa do jogo, não foi feita nenhuma interrupção à experiência do aluno, para que estes pudessem jogar o jogo e explorar as mecânicas sem intervenção externa. Para a segunda etapa, nos últimos dez minutos da aplicação, os alunos foram convidados a preencher o questionário de participação, disponível via Google Forms.

O público dos testes realizados identificou-se, portanto, como maioria (71,4%) na faixa etária entre 18 a 25 anos e que identifica a experiência própria com programação de média ou inferior.

Nas questões relativas à “Usabilidade e Confiança”, 60,7% dos respondentes declararam que foi fácil aprender a jogar o jogo e em relação à “As regras do jogo são claras e compreensíveis” 85,7% dos estudantes concordaram totalmente.

O conjunto de questões sobre “Desafio e Satisfação” mostrou que introduzir mecânicas e possíveis instruções para os personagens conforme o jogador entende as mecânicas já apresentadas em fases anteriores facilitou o entendimento do jogo, mantendo o desafio ao jogador. O terceiro conjunto de perguntas do questionário aborda diversão e atenção focada, buscando entender se o jogador se divertiu com o jogo, o que é importante para manter o jogador interessado no jogo educativo, e se o jogador se manteve focado no jogo durante o período de teste.

Em relação ao conjunto de perguntas sobre “Relevância e Aprendizagem” foi questionado se o jogo é relevante aos interesses do jogador, se é claro como o jogo está relacionado à disciplina e se o jogo contribui com a aprendizagem. As respostas a essas perguntas

indicam que os jogadores conseguiram entender a proposta do jogo e como ela se relaciona com o conteúdo estudado na disciplina.

A validação, aplicada por meio de questionários em turmas de Programação de Computadores I, proporcionou um bom feedback sobre o jogo como ferramenta de aprendizagem, além de apresentar novas ideias de como o jogo pode ser aplicado. O questionário finalizou com uma questão aberta sobre o feedback geral sobre a experiência com o jogo, por meio da qual pôde-se perceber uma boa aceitação por parte dos alunos, além de apresentar sugestões sobre novas mecânicas a serem exploradas. Como sugestões, destacam-se: criar um ranqueamento de acordo com um score gerado ao fim de cada fase, adicionar outros comandos relacionados à condicionais e possibilidade de alterar a velocidade das animações durante o combate.

5. Considerações Finais

A proposta inicial do trabalho, de planejar e desenvolver um jogo educativo com base nos métodos SGDA, para planejamento e desenvolvimento, e MEEGA+, para avaliação e validação, demonstrou-se possível. O jogo desenvolvido tem como principal objetivo o ensino de programação para estudantes de disciplinas iniciais no ensino superior, servindo como ferramenta de ensino, mas sem a intenção de substituir a aula, de forma com que o aluno possa relacionar e consolidar o aprendizado do conteúdo de estruturas condicionais em disciplinas de programação.

Ainda, de acordo com as aplicações de testes desenvolvidos com base no formulário do MEEGA+, o jogo demonstrou-se viável como ferramenta de ensino. Na aplicação dos questionários, com duas turmas de Programação de Computadores I, 28 alunos responderam, em grande parte, com um *feedback* bastante positivo sobre o jogo desenvolvido. O jogo foi baseado no conteúdo de estruturas condicionais, aplicada nas disciplinas de programação, porém pode também ser aplicado para currículos que incluem componentes de pensamento computacional, conforme os apresentados na seção 2. O conteúdo de condicionais encaixa nas disciplinas propostas pelo CIEB para currículos de 4º e 5º ano do ensino fundamental, desenvolvendo as dimensões curriculares de Algoritmos e de Reconhecimento de Padrões. Conforme as propostas de currículo do CIEB, essas dimensões propõem ao aluno executar algoritmos simples e identificar semelhanças e diferenças em situações que se repetem, para turmas de quarto ano; e conhecer e utilizar algoritmos com repetição e reconhecer padrões em um algoritmo, para turmas de quinto ano.

Além disso, nos *feedbacks* recebidos a partir do questionário, surgiram várias sugestões de melhorias para o jogo, que podem ser levadas a trabalhos futuros. Neste sentido, pode-se desenvolver mais disciplinas e mais conhecimentos relacionados a programação e pensamento computacional utilizando as mesmas mecânicas já existentes no jogo desenvolvido. O jogo também pode evoluir com novas fases, novas mecânicas, ou um sistema de ranqueamento de jogadores.

Referências

[1] Procura por profissionais de tecnologia cresce 671% durante a pandemia. CNN. 27 out. 2021, <https://www.cnnbrasil.com.br/business/procura-por-profissionais-de-tecnologia-cresce-671-durante-a-pandemia>

- [2] PETROPOULEAS, S. Nova leva de cursos tenta suprir aumento da demanda por profissionais de TI. Folha de S.Paulo. 1 dez. 2021, <https://www1.folha.uol.com.br/mercado/2021/12/demanda-por-profissionais-de-ti-supera-projecoes-e-impulsiona-iniciativas-para-suprir-deficit.shtml>
- [3] INSTITUTO SEMESP. In: Mapa do Ensino Superior. Cursos mais procurados. São Paulo, SP: Instituto SEMESP, 2021, <https://www.semesp.org.br/mapa/educacao-11/brasil/cursos-mais-procurados/>
- [4] INSTITUTO SEMESP. In: Mapa do Ensino Superior. Evasão. São Paulo, SP: Instituto SEMESP, 2021, <https://www.semesp.org.br/mapa/educacao-11/brasil/evasao/>
- [5] ABT, C. Serious Games. 1. ed. New York: Viking Press, 1970.
- [6] MITGUTSCH, K.; ALVARADO, N. Purposeful by design?: a serious game design assessment framework. FDG '12: Proceedings of the International Conference on the Foundations of Digital Games. New York: Association for Computing Machinery, 2012. p. 121-128.
- [7] PETRI, G.; WANGENHEIM, C.; BORGATTO, A. MEEGA+: Um Modelo para a Avaliação de Jogos Educacionais para o ensino de Computação. In: Revista Brasileira de Informática na Educação. Brazilian Journal of Computers in Education, 2019. p. 52-81.
- [8] SOUZA, D. M.; BATISTA, M. H.; BARBOSA, E. F. Problemas e Dificuldades no Ensino e na aprendizagem de Programação: Um Mapeamento Sistemático. Revista Brasileira de Informática na Educação. [S.l.], v. 24, n. 01, 2016. p 40-52.