

POOkémon: um jogo sobre programação orientada a objetos

Jaline Gonçalves Mombach* Bruno Morais Neves de Castro† Eduardo Schuindt Santos‡
Matheus Breno Batista dos Santos§

Instituto Federal de Educação, Ciência e Tecnologia de Brasília - IFB, Brasil



Figura 1: Tela inicial do jogo educativo POOKémon

RESUMO

Diversas pesquisas indicam a contribuição dos jogos digitais para fins de aprendizagem. Nas disciplinas de programação, em especial, os alunos podem transcender a posição de jogadores para também tornarem-se produtores de conteúdo ou idealizadores de jogos. Nesse sentido, este trabalho relata o desenvolvimento de POOKémon, um jogo educativo elaborado por alunos de ensino superior durante o curso da disciplina de Programação Orientada a Objetos. Codificado em linguagem Java, o jogo é baseado no clássico jogo de monstros, mas a habilidade de vencer do jogador está condicionada ao seu conhecimento acerca do conteúdo da disciplina. A metodologia de produção seguiu um modelo conhecido para desenvolvimento de jogos, contendo as fases de conceito, pré-projeto, protótipo, produção, alfa e beta. Realizaram-se testes com usuários para validação, e os resultados preliminares indicam que a proposta tem potencial educativo e motiva o aluno sobre a temática, além de personalização pelo professor.

Palavras-chave: Jogo Educativo, Programação, Java.

1 INTRODUÇÃO

Em meio a conquistas e avanços em áreas como Computação em Nuvem, Inteligência Artificial e Internet das Coisas, Cernuda [2] chama a atenção para o fato de atitudes simples de prática computacional ainda conseguirem promover mudanças relevantes em sala de aula. A maioria dessas iniciativas tem sido feita pela docência, objetivando uma melhor assimilação de conteúdo por parte dos estudantes e a criação de conteúdos que provoquem maior interatividade.

A utilização de jogos com fins educativos é considerada tendência em ambientes acadêmicos [3]. Com essa nova ferramenta, a capacidade de abstração do estudante é substancialmente

*e-mail: jaline.mombach@ifb.edu.br

†e-mail: brunomoraisnc@gmail.com

‡e-mail: eduschusan@gmail.com

§e-mail: mths.brn@gmail.com

explorada, tornando-se um objeto de aprendizagem participativo. Por meio da competição e dos desafios apresentados em jogo, é possível que o aprendiz desenvolva múltiplas inteligências e habilidades [1].

O panorama de jogos em disciplinas de informática atualmente compreende não somente a sua utilização como bem de consumo, mas, também, como matéria-prima que, quando alinhada à ementa disciplinar, torna-se capaz de trabalhar conceitos e desenvolver competências. Como exemplo, Rodrigues et al. [9] cita uma prática de ensino da disciplina de Programação Orientada a Objetos (POO) com a utilização do software Greenfoot, que compõe código em linguagem de programação Java a partir de interface gráfica para gerar um jogo, e o Robocode, responsável por criar robôs virtuais para batalha com o mesmo propósito.

A partir da perspectiva supracitada, este trabalho descreve o desenvolvimento de POOKémon, um jogo educativo elaborado por alunos de ensino superior durante o curso da disciplina de Programação Orientada a Objetos.

O artigo está organizado como segue. Na Seção 2, descreve-se as características de jogos considerados educacionais e na Seção 3, a metodologia de produção do jogo POOKémon. Na Seção 4, apresenta-se os resultados preliminares alcançados até a fase beta e por fim, na Seção 5 são discorridas as considerações finais.

2 JOGO DIGITAL EDUCACIONAL

Um *game* é uma atividade lúdica composta por uma série de ações e decisões que resultam em uma condição final [11]. Além disso, um jogo eletrônico define-se pela própria utilização de meios eletrônicos e programação digital em sua construção. Em comparação a jogos analógicos, a presença de regras e objetivos persiste, enquanto limitações são reduzidas e universos completos podem ser desenhados com grande complexidade gráfica e de enredo.

Segundo Piteira [8], jogos podem ser ferramentas capazes de proporcionar diversão enquanto motivam os alunos a aprenderem, facilitando-se o processo de aprendizado e a retenção do que lhes é ensinado. Adicionalmente, Kebritchi [5] afirma que jogos são ferramentas efetivas no ensino de processos complexos por enfatizarem a ação, em detrimento da explicação, e constituírem um ambiente que contextualize o processo decisório e a interação. Assim, com

a adição de elementos característicos, como combates, objetivos, regras e pontuação, é possível produzir recursos atrativos e de alto potencial didático.

Reforçando o modelo construtivista de ensino, centrado no aluno, a utilização de jogos na educação é comumente relacionada à teoria de aprendizagem experiencial de Kolb [6]. Nessa teoria, o estudante aprende por meio da reflexão sobre o processo de ação e interação com o objeto de estudo e o meio. Em conjunto, esses conceitos promovem a verticalização do ensino, em que o aluno aprende, cria e torna-se disseminador do conhecimento. Na perspectiva de estudo da computação, isso pode caracterizar um aprendizado por projeção e implementação de sistemas que, posteriormente, poderão ser utilizados como objetos de aprendizagem pelos próprios alunos e professores, implicando-se na autossuficiência de instrumentos didáticos complementares à formação.

Para Dempsey [4], um jogo instrucional deve: (1) ser relativamente simples de ser jogado, (2) ser adaptável e reprogramável de forma barata, (3) ter notório potencial para uso educacional quando adaptado, (4) ser diferente de outros jogos de sua categoria e (5) permitir jogo por um único jogador. Dessa forma, o jogo foi criado para se adequar a essas características, e o detalhamento de produção é apresentado na próxima Seção.

3 METODOLOGIA DE PRODUÇÃO

A teoria de Jeannie Novak [7] descreve as etapas essenciais para o desenvolvimento de jogos e apresenta um modelo com as etapas comumente adotadas em empresas de desenvolvimento. O processo formalizado pela autora compreende as fases exibidas na Figura 2.

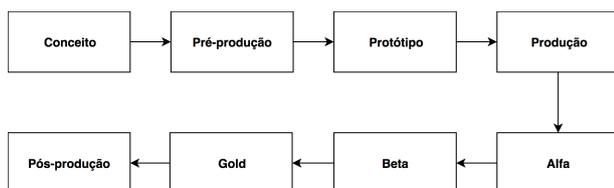


Figura 2: Modelo de Desenvolvimento de Jogos [7]

A fase de conceito é para discussão da ideia, identificar qual público-alvo, gênero e contexto do jogo. Na pré-produção são detalhadas regras, fases e personagens. Em seguida, são criados os protótipos, ou seja, as telas da interface e simulação de movimentos. Na produção, o jogo é implementado de fato, são desenhados os personagens e cenários, e programado em ambientes específicos.

As fases alfa, beta e gold compreendem a validação do produto em diferentes níveis, ou seja, na alfa o produto já pode ser jogado, mas a equipe analisa em busca de falhas. Na fase beta, o produto é distribuído para testes com usuários, e na ouro é encaminhado para fabricação. Por fim, na pós-produção o jogo já no mercado, mas recebe atualizações e expansões.

3.1 Conceito

O jogo POKémon surgiu por meio de uma atividade *brainstorming* em aula da disciplina de Programação Orientada a Objetos (POO). Inspirado pelo desenho Pokémon e pelos antigos jogos de *game boy* que fizeram muito sucesso na época de lançamento, foi notado um possível trocadilho envolvendo a disciplina de POO. Observada a dificuldade de memorização de determinados conceitos pela turma, a proposta foi definida como um jogo de perguntas e respostas, em que as quatro ações dos personagens, que apareciam durante as batalhas do jogo, seriam as respostas de questões referentes à própria disciplina. Portanto, perguntas deveriam ser aleatórias e alternativas sempre deveriam mudar de posição. O público-alvo são alunos que estejam cursando a disciplina, independente do nível, pois as perguntas são personalizadas pelo professor.

3.2 Pré-projeto

A linguagem de programação escolhida para implementação da plataforma foi Java, por ser a mais utilizada na disciplina de POO dos cursos do Instituto Federal de Brasília. Para composição da interface gráfica, utilizou-se o compositor de layout Gluon Scene Builder e JavaFX, uma plataforma desenvolvida pela Oracle para o desenvolvimento de aplicações que possam ser executadas em diferentes dispositivos. Ao ser combinada com o Gluon Scene Builder, permite a criação e implementação de aplicações de aparência moderna e conteúdo rico em animações, áudio e vídeo.

Para facilitar a atualização de conteúdo pela docência, foi estipulado que perguntas e respostas seriam armazenadas em um arquivo. Por ser o trabalho final da disciplina, exigiu-se, também, a utilização de elementos próprios de POO para modelagem e codificação do jogo, enfatizando a aprendizagem.

O arquivo para personalização do professor é uma planilha em formato *Comma-separated Values* (CSV) para alteração das questões. Consequentemente, é possível a sua utilização em diversos níveis, seja técnico ou superior. Contemplando os requisitos de jogo instrucional já apresentados, o POKémon se difere de outros jogos por ter sido produzido por alunos concluintes da disciplina, é de fácil interação por ser jogado com poucos cliques e apresenta partidas para um único jogador.

3.2.1 Regras

As regras definidas basearam-se em jogos comuns de perguntas e respostas. O jogador tem direito a escolher uma resposta por rodada. Caso a resposta escolhida esteja correta, a vida do monstinho oponente é reduzida, caso contrário, a vida do personagem do jogador é afetada. A partida finaliza quando a vida de um dos personagens for zerada. Um usuário pode capturar apenas um monstinho por partida.

3.2.2 Elementos visuais e interativos

Com o intuito de proporcionar nostalgia, chegou-se ao consenso de que o jogo teria os mesmos três personagens da versão de *game boy*: Bulbasaur, Squirtle e Charmander, onde, por exemplo, a nível de código, seria trabalhado o conceito de polimorfismo.

A partir das decisões elencadas, as seguintes telas da interface gráfica tornaram-se requisitos:

- Tela Inicial: é a tela de início do jogo, composta por um botão para inicializar o jogo.
- Tela de Escolha de Personagem: esta janela apresentará três personagens. O jogador deve poder escolher apenas um deles. Feita a escolha, um personagem oponente aleatório é criado.
- Tela da Partida: é a tela mais importante do jogo. Deve apresentar os personagens da tela anterior e suas respectivas barras de vida. Abaixo, o enunciado em uma caixa de texto e as respostas da rodada em botões. Quando escolhida uma resposta, a tela é atualizada com o conteúdo da pergunta seguinte.
- Tela de Erro de Resposta: é acionada quando o jogador errar uma pergunta. Deve conter o enunciado e seu gabarito.
- Tela de Pontuação: após o término da partida, apresenta a pontuação final do jogador, que, considerando ser um jogo educativo, não pode ser abaixo de zero. Deve apresentar um botão para voltar à tela inicial.

3.2.3 Definição de perguntas e respostas

Para validação, as perguntas e respostas foram resgatadas de um desafio aplicado em sala de aula. Mas, conforme já explicado, o professor pode personalizá-las como desejar.

3.3 Protótipo

O protótipo do jogo foi construído por meio de linguagem estruturada. Sendo de baixa complexidade, a linguagem estruturada não utiliza objetos e mais de uma classe em sua composição. Portanto, com a utilização da biblioteca *Swing*, foi possível testar os elementos visuais e interativos rapidamente.

3.4 Produção

A etapa de produção engloba o desenho e a organização das telas, a implementação do produto final e a modelagem do sistema, que neste trabalho representa a utilização dos conceitos de POO a nível de Análise e Projeto de Sistemas, outra disciplina básica de cursos de informática.

A diagramação foi construída em *Unified Modeling Language* (UML), por meio do utilitário *StarUML*, um modelador de código aberto. Esta arquitetura de diagramas de classe, casos de uso, atividade e sequência determinou como seria a estrutura, a interação e a integração entre as diversas partes da aplicação, tornando possível o prosseguimento da etapa de implementação.

O desenho de telas seguiu a técnica de design colaborativo. A composição foi executada tanto por meio de lápis e folhas de papel, quanto pela manipulação em *softwares* de tratamento de imagem, onde a utilização de guias para a aplicação da proporção áurea permitiu balancear os componentes de forma proporcionalmente agradável. Durante este processo, foi estabelecido que as perguntas e respostas seriam apresentadas com a questão à esquerda de quatro alternativas, e o espaço de batalha ficaria acima desses elementos.

É importante salientar que antes da estruturação de telas houve uma revisão dos componentes do Scene Builder. *TxtArea* e *Label* correspondem a elementos para representação de texto disponíveis no editor de leiaute.

3.4.1 Definição do arquivo de perguntas e respostas

Para inserção das perguntas no jogo, um arquivo CSV deve ser salvo no mesmo diretório que o arquivo Java executável (.jar) com o nome “perguntas.csv”. Para criar o arquivo, é necessário um editor de texto simples, como o Bloco de Notas do *Windows* ou *Kwrite* do *Linux*. O caractere separador é o ponto e vírgula (;) e, por isso, permite a inserção de caracteres especiais no enunciado e nas alternativas. As questões e possíveis respostas não devem conter o separador. O arquivo deve possuir codificação UTF-8 para leitura dos acentos e pontuação de maneira adequada. A Figura 3 exibe um exemplo de planilha para personalização pelos professores.

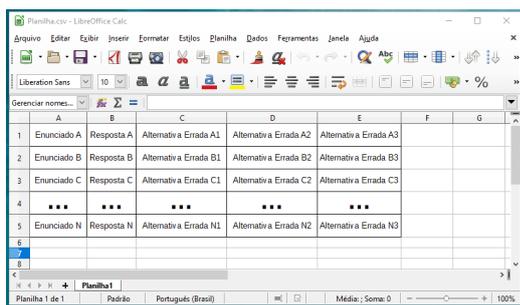


Figura 3: Arquivo de Planilha em edição no LibreOffice Calc

Cada linha representa uma pergunta a ser apresentada no jogo. O número mínimo de perguntas é 4, não havendo um máximo. Se o arquivo comportar entre 4 a 15 perguntas, o jogo usará todas as perguntas inseridas. Caso exceda este valor, o jogo seleciona 15 perguntas aleatoriamente a cada nova partida. Apesar de não existir um limite de perguntas, o ideal é que o arquivo de perguntas tenha o menor tamanho possível, pois, após a leitura, as perguntas ocupam

uma parte da memória RAM do computador quando em tempo de execução.

3.5 Fases Alfa e Beta

A escolha da *JavaFX* como *framework* permitiu que os testes da interface gráfica fossem realizados em navegadores modernos como o *Mozilla Firefox*, *Google Chrome* e *Microsoft Edge*. As telas projetadas para o jogo apresentaram consistência de cores, elementos e comportamentos em todos os testes.

Na versão alfa, a lógica do jogo foi testada através de saídas exibidas no terminal durante a execução do jogo. A cada execução, os valores das perguntas selecionadas e das sequências de alternativas foram anotadas e comparadas com o arquivo de perguntas para averiguar se eram selecionadas e exibidas aleatoriamente. Apesar dos vários testes realizados, não foram encontrados casos de repetição das ordens.

Os valores apresentados indicando a barra de vida e a pontuação também eram exibidos no terminal enquanto o jogo era executado. Como ferramenta de controle, os cálculos a cada acerto ou erro e a pontuação final foram feitos paralelamente em um *software* de planilha. Os resultados obtidos foram os mesmos em ambos. A Figura 4 mostra a tela de escolha de personagem do jogo e a Figura 5 a simulação de batalha contra o computador.



Figura 4: Tela de escolha de personagem

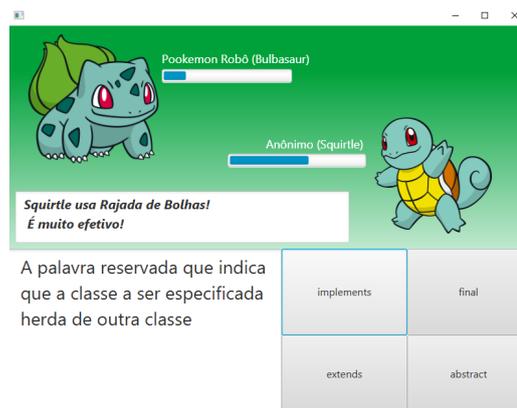


Figura 5: Tela de batalha

3.5.1 Validação com Usuários

Na versão beta, solicitou-se a estudantes da disciplina de POO que testassem o jogo e, em seguida, respondessem um questionário de

avaliação. Para realização do procedimento de validação, foi utilizado o modelo de avaliação de jogos educacionais de Savi [10], por priorizar a objetividade das perguntas e respostas. Participaram 6 alunos, sendo 3 de curso técnico integrado ao ensino médio e 3 de curso superior. Os resultados são apresentados na próxima Seção.

4 RESULTADOS PRELIMINARES

O jogo atualmente está na versão beta, ou seja, segundo o modelo de [7] seria necessário encaminhamentos para a versão gold e a fase de pós-produção. Porém, salienta-se que o jogo tem contexto educativo, e não comercial, e, por isso, teve esse modelo adaptado e apresenta resultados preliminares positivos, mesmo na fase beta. Adaptou-se o modelo de [10], contemplando as categorias para avaliação quanto à Motivação (atenção, relevância, confiança e satisfação) e Experiência do Usuário (Imersão, Desafio, Habilidade/Competência) com escala Likert de 5 pontos: discordo fortemente, discordo, não concordo nem discordo, concordo e concordo fortemente. Além deste, um campo aberto para sugestões de melhoria foi colocado no questionário.

A avaliação com usuários obteve resultados positivos, dado que mais da metade dos estudantes que responderam o formulário concordaram que o *design* da interface do jogo é atraente e todos concordaram com a afirmativa: “o jogo é tão interessante que gostaria de aprender mais sobre o assunto abordado por ele”. Ainda, os estudantes marcaram que não houve dificuldade para entender o jogo, conseguiram associar com conteúdos já aprendidos e que o nível de complexidade estava adequado, conforme exibido na Figura 6. Também, a maioria concordou que aprendeu conceitos surpreendentes ou inesperados, conforme o gráfico da Figura 7.

- Este jogo é adequadamente desafiador para mim, as tarefas não são muito fáceis nem muito difíceis.

6 responses

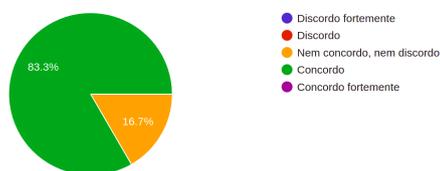


Figura 6: Gráfico sobre adequação do desafio.

- Eu aprendi algumas coisas com o jogo que foram surpreendentes ou inesperadas.

6 responses

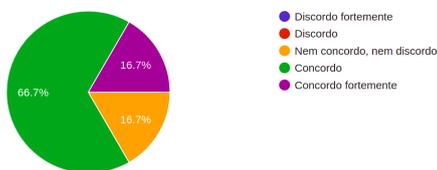


Figura 7: Gráfico sobre aprendizagem surpreendente.

Por outro lado, os participantes sugeriram melhorias, como a implementação de uma trilha sonora, sistema de recompensas nas fases e ter mais questões. Ao aplicar os testes, os alunos (desenvolvedores) perceberam também necessidade de melhoras quanto à usabilidade, como contraste de cores, redimensionamento da tela e opção mais intuitiva para dar seguimento ao jogo após errar. Con-

forme ilustrado na Figura 8, atualmente, ao errar uma questão, o jogador deve fechar a janela para continuar jogando.



Figura 8: Tela de erro de resposta

5 CONSIDERAÇÕES FINAIS

Este trabalho apresenta o desenvolvimento de um jogo por alunos do ensino superior durante realização da disciplina de POO. O diferencial da proposta é promover aos alunos a possibilidade de aplicar os conhecimentos da disciplina na implementação do jogo (programação em Java com JavaFX) e também a parte conceitual de POO na elaboração de perguntas e respostas. Além disso, ao elaborar um jogo conforme as fases usuais de desenvolvimento, ou seja, conceituação, produção e testes, os alunos experimentam na prática o processo e podem replicar o que foi aprendido. O projeto obteve bons resultados para os alunos desenvolvedores e também nos testes realizados com outros estudantes, ou seja, os usuários.

Os trabalhos futuros indicam a disponibilização do jogo em git, além de publicação on-line para uso como objeto de aprendizagem por professores da área de programação.

REFERÊNCIAS

- [1] C. Antunes. *Jogos para a estimulação das múltiplas inteligências*. Editora Vozes Limitada, 2011.
- [2] C. Cernuda. *A Transformação Digital feita por professores*. Microsoft News Center Brasil, April 2017.
- [3] S. de Sena and A. H. Catapan. Metodologias para a criação de jogos educativos: uma revisão sistemática da literatura. *Renote*, 14(2), 2016.
- [4] J. V. Dempsey et al. *Instructional applications of computer games*. 1996.
- [5] M. Kebritchi and A. Hirumi. Examining the pedagogical foundations of modern educational computer games. *Computers & Education*, 51(4):1729 – 1743, 2008.
- [6] D. A. Kolb. *Experiential learning: Experience as the source of learning and development*. FT press, 2014.
- [7] J. Novak. *Desenvolvimento de games*. São Paulo: Cengage Learning, pages 354–355, 2010.
- [8] M. Piteira and S. R. Haddad. Innovate in your program computer class: An approach based on a serious game. In *Proceedings of the 2011 Workshop on Open Source and Design of Communication, OSDOC '11*, pages 49–54, New York, NY, USA, 2011. ACM.
- [9] L. C. Rodrigues, G. C. Nogueira, and A. Queiroga. Experiências no ensino de programação orientada a objetos: RoboCode, greenfoot e jogos de tabuleiro no ensino superior. In *Anais do XXIII Workshop de Informática na Escola (WIE 2017)*. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC), oct 2017.
- [10] R. Savi, C. G. Von Wangenheim, V. Ulbricht, and T. Vanzin. Proposta de um modelo de avaliação de jogos educacionais. *Renote*, 8(3), 2010.
- [11] P. Schuytema. *Design de games: uma abordagem prática*. CENGAGE DO BRASIL, 2008.