Fractália: Uma Abordagem Lúdica para o Ensino do Pipeline de Visão Computacional

Naila Suele Mascarenhas Silva[1], **Victor Travassos Sarinho**[1]

[1]Universidade Estadual de Feira de Santana (UEFS)
Feira de Santana – BA – Brasil

nailasuely@ecomp.uefs.br, victor.sarinho@uefs.br

Abstract. *Literacy in Machine Learning (ML) and Computer Vision (CV) is increasingly relevant in the 21st century, yet introductory teaching faces barriers due to mathematical abstraction and the complexity of the data pipeline. This work presents Fractália, a serious game developed in the Godot engine that introduces the CV data pipeline through active learning. The game translates technical concepts into gameplay mechanics, such as managing computational energy and recognising patterns, allowing players to experience the stages of preprocessing, annotation (Ground Truth), convolution and inference evaluation through metaphors. This is a development report: the article presents the design decisions and the implemented mechanics that express the sequential interdependence of the pipeline, and outlines the user evaluation protocol planned as future work, offering a complementary tool for computing education. The results reported concern technical and design feasibility, not proven pedagogical effectiveness, which depends on the planned user evaluation.*

Resumo. *O letramento em Aprendizado de Máquina (ML) e Visão Computacional (VC) é cada vez mais relevante no século XXI, mas o ensino introdutório enfrenta barreiras decorrentes da abstração matemática e da complexidade do pipeline de dados. Este trabalho apresenta o Fractália, um jogo sério desenvolvido na engine Godot que introduz o pipeline de dados de VC por meio de aprendizagem ativa. O jogo traduz conceitos técnicos em mecânicas de jogabilidade, como o gerenciamento de energia computacional e o reconhecimento de padrões, permitindo que os jogadores vivenciem as etapas de pré-processamento, anotação (Ground Truth), convolução e avaliação da inferência por meio de metáforas. Trata-se de um relato de desenvolvimento: o artigo apresenta as decisões de design e as mecânicas implementadas que expressam a interdependência sequencial do pipeline, e descreve o protocolo de avaliação com usuários planejado como trabalho futuro, oferecendo uma ferramenta complementar para o ensino de computação. Os resultados relatados dizem respeito à viabilidade técnica e de design, e não à eficácia pedagógica comprovada, que depende da avaliação com usuários prevista.*

1. Introdução

A crescente adoção de sistemas baseados em Aprendizado de Máquina (ML) e Visão Computacional (VC) tem ampliado a procura por estratégias pedagógicas que aproximem estudantes de computação da prática da área. O ensino tradicional frequentemente

prioriza fundamentos matemáticos densos antes de apresentar contextos práticos, o que pode gerar carga cognitiva extrínseca e dificultar a compreensão do pipeline de desenvolvimento [Sweller 2011].

Um desafio central reside na desconexão entre teoria algorítmica e prática de engenharia de dados. Estudantes iniciantes tendem a focar na otimização de hiperparâmetros e a negligenciar etapas fundamentais como a limpeza de dados e a criação de bases de verdade (*Ground Truth*), responsáveis pela maior parte do esforço em projetos reais [Sambasivan et al. 2021]. Em uma disciplina introdutória, esse desequilíbrio se manifesta, por exemplo, quando o aluno atribui o baixo desempenho de um classificador exclusivamente à configuração do modelo, sem perceber que rótulos inconsistentes ou ruído no conjunto de treinamento limitam o resultado independentemente do ajuste fino realizado.

Para mitigar essa lacuna educacional, a Aprendizagem Baseada em Jogos Digitais (*Digital Game-Based Learning*, DGBL) surge como alternativa para situar o aprendizado em contextos significativos [Prensky 2003]. É importante distinguir conceitos próximos, porém distintos. A gamificação consiste no uso de elementos de jogos em contextos que não são jogos, como processos administrativos ou educacionais [Deterding et al. 2011]; já um jogo sério é um jogo completo cujo propósito vai além do entretenimento, incorporando objetivos de aprendizagem na própria estrutura [Hakulinen 2011]. O Fractália pertence a esta segunda categoria: é um jogo sério, projetado para que a compreensão dos conceitos de VC emerja da própria jogabilidade.

Neste sentido, este trabalho apresenta o design e o desenvolvimento do Fractália, um jogo sério que traduz o pipeline de dados de VC em mecânicas de simulação e gerenciamento de recursos. Como relato de desenvolvimento, suas contribuições são as decisões de projeto, a implementação da interdependência entre as etapas do pipeline como mecânica jogável e o protocolo de avaliação com usuários previsto. O jogo é concebido como ferramenta complementar, e não como substituto do ensino formal e matemático, que permanece indispensável à área.

2. Fundamentação Teórica

A Visão Computacional é definida por Szeliski [Szeliski 2022] como a ciência que busca inferir descrições do mundo a partir de imagens ou sequências de imagens. O autor enfatiza que, embora as abordagens modernas sejam dominadas pelo Aprendizado Profundo (*Deep Learning*), o entendimento dos princípios fundamentais, tais como a formação da imagem, o pré-processamento para remoção de ruído e a extração de características, permanece vital. O ciclo de vida típico de um sistema de visão envolve a aquisição de dados, o pré-processamento (limpeza), a anotação (para aprendizado supervisionado), o treinamento do modelo e a inferência [Szeliski 2022, Goodfellow et al. 2016]. Em contextos educacionais, facilitar o entendimento da matemática complexa por trás da convolução ou da retropropagação (*backpropagation*), sem perder a precisão conceitual, é um desafio que jogos de simulação podem ajudar a enfrentar [Gee 2007].

O uso de jogos sérios no ensino de algoritmos e programação tem sido investigado na literatura de educação em computação [Hakulinen 2011, Gee 2007]. Hakulinen [Hakulinen 2011] discute como jogos podem ser utilizados para ensinar estruturas de dados e algoritmos, argumentando que a interação ativa e a discussão geradas pe-

las mecânicas de jogo tendem a favorecer o engajamento em relação a métodos passivos. Para o autor, um jogo é eficaz quando a mecânica principal está alinhada ao objetivo de aprendizagem, evitando que o conteúdo educacional seja apenas um adendo superficial à diversão. Essa perspectiva se alinha ao conceito de integração intrínseca [Habgood and Ainsworth 2011], em que a mecânica do jogo é, por si só, o conteúdo a ser aprendido.

3. Trabalhos Relacionados

Especificamente no domínio da Inteligência Artificial, Frutos-Pascual e García-Zapirain [Frutos-Pascual and García-Zapirain 2017] realizaram uma revisão sistemática sobre o uso de técnicas de IA em jogos sérios. O estudo aponta que, embora a IA seja frequentemente empregada para controlar personagens não jogáveis (NPCs) ou adaptar a dificuldade, existe escassez de jogos cujo conteúdo instrucional seja a própria engenharia de sistemas inteligentes. Iniciativas anteriores ilustram caminhos distintos para esse ensino. No contexto brasileiro, Scardua e Develly [Scardua and Develly 2021] aplicaram a gamificação ao ensino de Inteligência Artificial na engenharia de controle e automação, com ênfase em elementos motivacionais externos ao conteúdo. Em escala mais ampla, ambientes de treino de agentes em que o jogador ajusta hiperparâmetros e observa o comportamento resultante colocam a IA como objeto de manipulação, embora ainda abordem com pouca profundidade as etapas de curadoria de dados que antecedem o treinamento. O Fractália distingue-se ao tratar justamente o trabalho de dados como mecânica central, e não como pré-requisito invisível. A Tabela 1 situa essa diferença em relação às abordagens mencionadas.

Tabela 1. Posicionamento do Fractália frente a abordagens correlatas de ensino de IA.

Abordagem	**Trabalho de dados como mecânica**	**Pipeline completo**	**Integração intrínseca**
Gamificação do ensino de IA (Scardua e Develly)	Não	Não	Não (elementos externos)
Ambientes de treino de agentes / ajuste de hiperparâmetros	Não	Parcial (foco no modelo)	Parcial
Fractália (este trabalho)	**Sim**	**Sim**	**Sim**

Neste contexto, o Fractália busca preencher essa lacuna ao colocar o jogador no papel de um engenheiro de ML, e não apenas de um usuário de sistemas prontos. Em vez de abstrair a complexidade, o jogo a simplifica por meio de metáforas visuais, permitindo que o jogador compreenda a lógica de sistemas complexos sem a necessidade imediata de implementação de código.

4. Metodologia de Desenvolvimento

O desenvolvimento do Fractália seguiu uma abordagem iterativa de design centrada no alinhamento entre mecânica e objetivo de aprendizagem. Três princípios orientaram as decisões de projeto. O primeiro foi a integração intrínseca [Habgood and Ainsworth 2011]: cada fase foi concebida de modo que a ação exigida do jogador correspondesse diretamente a uma etapa do pipeline de VC, evitando que o conteúdo técnico fosse apenas justaposto à jogabilidade. O segundo foi a redução da carga cognitiva extrínseca [Sweller 2011], obtida pela centralização do estado do jogo em um núcleo lógico único e pela apresentação da informação no momento de uso. O terceiro foi a busca por decisão significativa [Salen and Zimmerman 2004], segundo a qual a escolha do jogador só tem valor quando produz um resultado discernível e integrado ao contexto maior do jogo.

A arquitetura foi organizada de forma modular, com cada etapa do pipeline representada por uma cena independente, decisão que favorece a manutenção e a inclusão de novas fases. O estado compartilhado entre as fases (recursos, métricas e progresso do modelo) é mantido por um único componente global, o que garante consistência e desacopla a interface da lógica de simulação. O jogo foi implementado na versão 4.6 da engine de código aberto Godot [Linietsky and Manzur 2014], escolha coerente com os valores de reprodutibilidade da educação em computação e que facilita a distribuição do artefato.

5. Jogo Produzido

Fractália (Figura 1) adota a arquitetura modular descrita na seção anterior. Um script de gerenciamento global atua como núcleo lógico do sistema e centraliza as variáveis compartilhadas entre as fases, o que garante a consistência do estado da aplicação. Esse componente administra quatro recursos principais que, como simplificações lúdicas, representam restrições típicas de um ambiente real de desenvolvimento de modelos de Machine Learning:

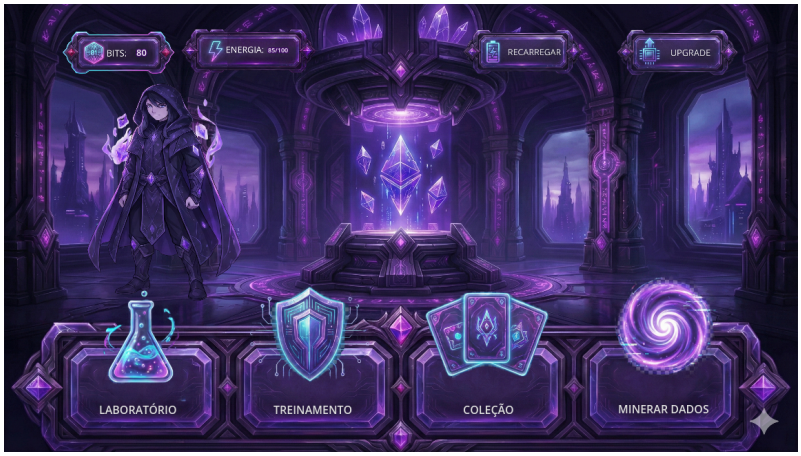


Figura 1. Hub central do Fractália: gerenciamento de recursos (Bits e Energia) e acesso aos módulos do pipeline.

A **energia** é uma metáfora simplificada do custo computacional associado à GPU, sem corresponder a uma medida técnica exata de capacidade, latência ou vazão. É consumida por todas as operações de processamento de dados, sejam elas de mineração, anotação ou treinamento. O sistema impõe um limite máximo, o que obriga o jogador a

realizar pausas estratégicas para recarga e transmite a ideia de que o treinamento de modelos não é instantâneo e possui custos computacionais significativos. Os **bits** funcionam como a moeda do jogo, obtida por meio do sucesso nas missões, e são empregados na aquisição de atualizações de hardware, como o aumento da capacidade de energia, que permite sessões de treinamento mais longas e introduz uma camada de gestão estratégica ao projeto. Os **pacotes de dados** representam o insumo bruto. O jogador não avança para a fase de anotação sem antes ter minerado pacotes suficientes, o que estabelece uma dependência sequencial entre as etapas do pipeline. Por fim, o **nível de IA** e a **acurácia** determinam a eficácia do modelo, são atualizados com base no desempenho do jogador nos minigames e influenciam diretamente o feedback visual na galeria de modelos.

Do ponto de vista do jogador, o objetivo geral do Fractália é elevar a acurácia do modelo até a meta estabelecida, conduzindo o pipeline da matéria-prima ao modelo estável. Cada fase concede recompensas (Bits e ganho de acurácia) quando bem executada, e a partida é vencida quando a acurácia global atinge o limiar de qualidade. O jogador é desafiado a cada fase por restrições de tempo, de energia e de qualidade dos dados; falhas não encerram a partida, mas reduzem o ganho e exigem novas iterações, refletindo o caráter incremental do trabalho real com dados. O público-alvo são estudantes de graduação em cursos de computação e áreas correlatas, em fase introdutória ao tema, embora a mediação por metáforas torne o jogo acessível também a interessados sem formação prévia em ML.

5.1. Mapeamento entre fases, mecânicas e conceitos

A principal contribuição do jogo reside na tradução de conceitos abstratos de VC em mecânicas lúdicas concretas e interativas. A Tabela 2 apresenta a correspondência sistemática entre as quatro fases do jogo, suas mecânicas de *gameplay* e os conceitos técnicos de VC/ML que representam.

5.2. Interdependência do pipeline como mecânica

O Fractália foi projetado para que a interdependência sequencial do pipeline seja percebida durante a jogabilidade, e não apenas descrita textualmente. Para isso, as fases foram conectadas pelo estado global do modelo. A limpeza incompleta na Mineração deixa um ruído residual que reduz o teto de acurácia alcançável no Treinamento e aumenta a proporção de anomalias que o jogador precisa filtrar. De modo análogo, a qualidade da anotação produzida no Laboratório define o teto de acurácia que o Treinamento pode atingir, pois o aprendizado supervisionado depende de rótulos consistentes. O resultado do Treinamento, por sua vez, alimenta a acurácia global avaliada na Coleção. Essa cadeia traduz em jogo a ideia, discutida por Sambasivan et al. [Sambasivan et al. 2021], de que a subvalorização do trabalho de dados gera efeitos que se acumulam ao longo do pipeline e limitam o modelo.

Na fase de Mineração, o ruído se propaga com o tempo: a cada intervalo, um bloco corrompido pode contaminar um vizinho íntegro. O jogador pode compilar o conjunto de dados a qualquer momento, o que introduz uma decisão significativa [Schell 2008]: compilar cedo, com um conjunto menor e mais seguro, ou limpar por mais tempo em busca de maior recompensa, sob o risco de a contaminação se espalhar. A fase expressa o compromisso entre remover anomalias e perder informação, pois remover blocos íntegros também penaliza o resultado.

Tabela 2. Mapeamento entre fases, mecânicas e conceitos de VC/ML.

Fase do jogo	Mecânica de *gameplay*	Conceito de VC/ML ensinado
Mineração	Inspeção visual e eliminação de blocos corrompidos da grade antes da compilação do conjunto de dados (Figura 2a).	Pré-processamento e limpeza de dados; impacto da propagação de ruídos no modelo; balanço entre exclusão de anomalias e perda de informação.
Treinamento	Movimentação de uma torre (filtro) para bloquear anomalias e permitir a passagem de pacotes válidos (Figura 2b).	Aplicação de filtros de convolução na extração de características; ajuste de pesos; tratamento de anomalias (*outliers*); convergência.
Laboratório	Demarcação manual de áreas de interesse clicando sobre alvos ocultos na imagem (Figura 3).	Aprendizado supervisionado; anotação de dados e criação da verdade fundamental (*Ground Truth*); dependência de dados consistentes.
Coleção	Visualização dos resultados com variação paramétrica de opacidade e cor baseada no nível do modelo (Figura 4).	Avaliação de inferência na galeria; leitura visual da acurácia; identificação de subajuste frente ao modelo estabilizado.

No Laboratório (Figura 3), o jogador dispõe de tempo limitado para demarcar manualmente os alvos de interesse, e cada marcação imprecisa é contabilizada como rótulo de baixa qualidade. A precisão da anotação vira ganho real de dados rotulados, o que estabelece a decisão entre anotar com rapidez ou com cuidado.

Na fase de Coleção, o estado do modelo é comunicado de forma visual, sem recorrer a texto. Um modelo subajustado apresenta baixa fidelidade e instabilidade, enquanto um modelo estável é exibido com nitidez, conforme o contraste evidenciado na Figura 4. A fidelidade visual da reconstrução é mapeada a partir da acurácia do modelo, de modo que a leitura do desempenho não dependa apenas do valor numérico. Essa redundância entre forma e número segue o princípio multimídia da apresentação por múltiplos canais [Mayer 2009], permitindo que o jogador identifique visualmente a diferença entre subajuste e o modelo estabilizado.

5.3. Recursos, economia e mentoria

A condição de vitória confere à sequência de ações um objetivo global: ao atingir a meta de acurácia, o jogador vence a partida. Para sustentar o ritmo, a economia de Bits compõe o ciclo e permite recarregar a energia ou ampliar a capacidade de GPU. Esse controle sobre o próprio ritmo de jogo atua como fator de motivação intrínseca [Malone and Lepper 1987]. A calibragem da dificuldade segue a teoria do fluxo [Csikszentmihalyi 1990], com desafio mantido ligeiramente acima da habilidade do jogador e crescente conforme o nível do modelo avança.

Um sistema de mentor virtual, representado pela personagem Fei (Figura 5), ajuda o jogador a compreender os conceitos técnicos sem longos textos expositivos. O mentor cumpre dois papéis: explica cada mecânica na estreia da fase, para evitar repetição, e

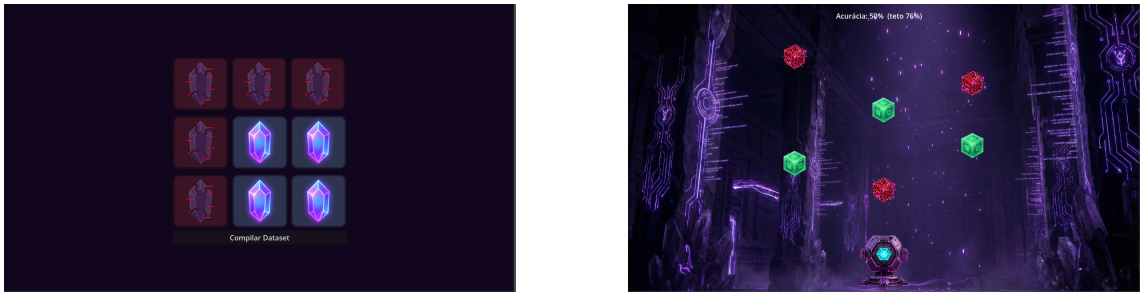


Figura 2. Fases iniciais: (a) remoção de blocos de ruído no módulo de Mineração e (b) torre-filtro bloqueando ruído (vermelho) e deixando passar dados válidos (verde) na fase de Treinamento.

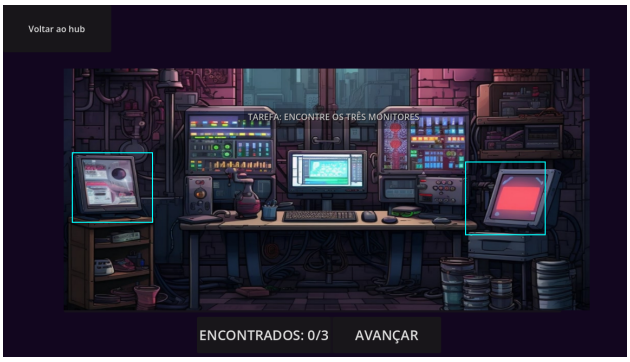


Figura 3. Módulo de Laboratório: demarcação manual de áreas de interesse para a construção da verdade fundamental (*Ground Truth*).

orienta o próximo passo ao término de cada ação, com base no estado real do pipeline. Essa abordagem segue o princípio de aprendizagem *Just-in-Time* [Gee 2007], em que a informação aparece no momento exato em que se torna necessária para a ação.

6. Validação Técnica e Protocolo de Avaliação Previsto

A implementação foi validada tecnicamente por execução automatizada de cada cena, que instancia os módulos e exercita os laços de simulação (temporizadores, propagação de ruído e atualização de recursos), verificando a ausência de erros e a corretude da cadeia de interdependência ao longo de um ciclo completo do pipeline. Cabe enfatizar que essa validação confirma a viabilidade técnica e a coerência da lógica, mas não constitui avaliação pedagógica: não mede a experiência de jogo nem o ganho de aprendizagem.

A avaliação com usuários está planejada como trabalho futuro e seguirá um protocolo já definido. Para a qualidade percebida do jogo, o protocolo utilizará o modelo MEEGA+ [Petri et al. 2019], instrumento validado na comunidade de educação em computação para avaliar motivação, experiência do usuário e percepção de aprendizagem. Para o ganho de conteúdo, o estudo aplicará um pré e pós-teste curto sobre as etapas do pipeline (limpeza, anotação, convolução e avaliação), com questões estruturadas segundo a Taxonomia de Bloom [Anderson and Krathwohl 2001], para distinguir níveis de aprendizagem, da lembrança de conceitos à sua aplicação. O registro de interação coletado durante a sessão (acurácia final atingida e número de iterações por fase) servirá de medida objetiva de desempenho. A amostra inicial prevista é de cerca de 15 a 30 estudantes de graduação em fase introdutória ao tema, recrutados preferencialmente a partir de tur-

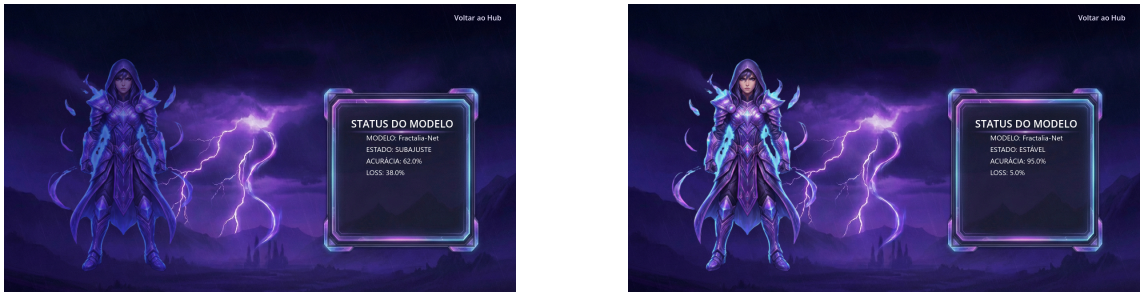


Figura 4. Avaliação de inferência na Coleção: (a) modelo subajustado, com baixa acurácia e fidelidade visual reduzida, e (b) modelo estável, com alta acurácia e reconstrução nítida.



Figura 5. A mentora virtual Fei orientando o jogador na fase de Mineração, com diálogo apresentado no momento da ação (princípio *Just-in-Time*).

mas completas de disciplinas introdutórias, e não apenas de voluntários, a fim de reduzir o viés de autosseleção. Como limitação de desenho, esta primeira etapa não contará com grupo de controle, o que impede atribuir ao jogo, de forma isolada, eventuais ganhos observados; será, portanto, um estudo exploratório, cujos resultados serão tratados como preliminares. Uma etapa posterior prevê a comparação com um grupo de controle submetido a uma aula expositiva tradicional, o que permitirá aplicar testes de significância, como o de Wilcoxon, e medidas de tamanho de efeito, como o d de Cohen, para fortalecer a inferência sobre a aprendizagem.

7. Considerações Finais

O desenvolvimento do Fractália indica a viabilidade técnica e o potencial pedagógico de traduzir o fluxo de trabalho de Visão Computacional para um ambiente de jogo acessível. A arquitetura implementada estabelece uma estrutura em que o sucesso no treinamento depende da qualidade da anotação, que por sua vez depende da limpeza correta dos dados brutos. Essa interdependência, antes apenas descrita, passou a ser exercida pela própria jogabilidade, com o objetivo de reforçar uma ideia relevante para o letramento em IA: a de que a qualidade dos dados condiciona o desempenho do modelo. Com isso, o jogo busca contribuir para desmistificar a noção de que a IA é uma caixa preta que funciona independentemente da intervenção humana.

Reconhecem-se, contudo, limitações que delimitam o escopo deste trabalho. A validação conduzida até aqui é de natureza técnica, e não pedagógica: confirma que o sistema funciona como projetado, mas não evidencia ganho de aprendizagem. As afirmações sobre o valor educacional do jogo se apoiam no design e na literatura, e não em dados empíricos com usuários. A avaliação prevista, por sua vez, é exploratória e não contará, em sua primeira etapa, com grupo de controle. Além disso, os recursos do jogo são simplificações lúdicas e não correspondem a medidas técnicas exatas dos fenômenos que representam, o que introduz um risco pedagógico: um estudante pode, por exemplo, interpretar a energia como uma medida real de latência ou vazão da GPU. Cabe à mentora Fei mitigar essas leituras equivocadas, ao explicitar o caráter metafórico dos recursos no momento em que são introduzidos[1]. O escopo atual também se concentra em estudantes iniciantes; a adequação do jogo a outros públicos, como estudantes de ensino médio ou profissionais em requalificação, permanece uma questão em aberto.

Diante desse cenário, os próximos passos buscam superar essas limitações. Além da avaliação com usuários, o projeto prevê a criação de novas fases que introduzam o conceito de sobreajuste (*overfitting*). Em um esboço conceitual dessa mecânica, o jogador seria recompensado por ajustar o modelo repetidamente aos mesmos dados já vistos, elevando a acurácia no conjunto de treino; ao avançar para a Coleção, porém, o modelo seria confrontado com imagens inéditas e revelaria baixo desempenho, tornando visível o contraste entre memorização e generalização. O projeto pretende, ainda, expandir as fases para abordar outros tópicos de Visão Computacional e Aprendizado de Máquina, aprofundando os conceitos explorados ao longo do jogo.

Declaração sobre o uso de Inteligência Artificial Generativa

Em conformidade com o Código de Conduta da SBC, declara-se que ferramentas de Inteligência Artificial Generativa foram utilizadas exclusivamente como apoio à revisão textual e à verificação de clareza da redação. A concepção, o design, a implementação do jogo, a análise e as conclusões deste trabalho são de autoria integral dos autores.

Referências

Anderson, L. W. and Krathwohl, D. R. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Longman, New York.

Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. Harper and Row, New York.

Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011). From game design elements to gamefulness: Defining "gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pages 9–15, Tampere, Finland. ACM.

Frutos-Pascual, M. and García-Zapirain, B. (2017). Review of the use of ai techniques in serious games: Decision making and machine learning. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(2):133–152.

[1]Na fase de Mineração, ao consumir energia, a mentora Fei esclarece que o recurso representa o custo computacional de forma abstrata, e não uma métrica técnica exata.

Gee, J. P. (2007). *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan, New York, 2nd edition.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, MA.

Habgood, M. P. J. and Ainsworth, S. E. (2011). Motivating children to learn effectively: Exploring the value of intrinsic integration in educational games. *Journal of the Learning Sciences*, 20(2):169–206.

Hakulinen, L. (2011). Using serious games in computer science education. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, pages 83–88, Koli, Finland. ACM.

Linietsky, J. and Manzur, A. (2014). Godot engine. Open-source game engine. Disponível em: https://godotengine.org.

Malone, T. W. and Lepper, M. R. (1987). Making learning fun: A taxonomy of intrinsic motivations for learning. In Snow, R. E. and Farr, M. J., editors, *Aptitude, Learning, and Instruction*, volume 3, pages 223–253. Lawrence Erlbaum Associates, Hillsdale, NJ.

Mayer, R. E. (2009). *Multimedia Learning*. Cambridge University Press, Cambridge, 2nd edition.

Petri, G., von Wangenheim, C. G., and Borgatto, A. F. (2019). MEEGA+: Um modelo para a avaliação de jogos educacionais para o ensino de computação. *Revista Brasileira de Informática na Educação*, 27(3):52–81.

Prensky, M. (2003). Digital game-based learning. *Computers in entertainment (CIE)*, 1(1):21–21.

Salen, K. and Zimmerman, E. (2004). *Rules of Play: Game Design Fundamentals*. MIT Press, Cambridge, MA.

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., and Aroyo, L. M. (2021). "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Scardua, L. A. and Develly, D. P. (2021). Gamificação no ensino de inteligência artificial aplicada à engenharia de controle e automação. *Revista Científica*. Instituto Federal do Espírito Santo (Ifes).

Schell, J. (2008). *The Art of Game Design: A book of lenses*. CRC press.

Sweller, J. (2011). Cognitive load theory. In *Psychology of learning and motivation*, volume 55, pages 37–76. Elsevier.

Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer, Cham, 2nd edition.