

USO DO GODOT PARA CRIAÇÃO DE SIMULADORES GAMIFICADOS

 <https://doi.org/10.56238/sevened2024.041-035>

Daniel de Andrade Moura

Professor Doutor de física
Instituto Federal de São Paulo, São Paulo, SP
dmoura@ifsp.edu.br

Sabrina Souza da Silva

Licencianda em Física
Instituto Federal de São Paulo, São Paulo, SP
silva.sabrina@aluno.ifsp.edu.br

Lucas Martins

Licenciando em Física
Instituto Federal de São Paulo, São Paulo, SP
18martins.lucas@gmail.com

Marina Mendes Rezende

Graduanda em Engenharia de controle e automação
Instituto Federal de São Paulo, São Paulo, SP
marina.rezende@aluno.ifsp.edu.br

Guilherme Ferreira da Silva

Licenciando em Química
Instituto Federal de São Paulo, São Paulo, SP
guilherme.ferreira3@aluno.ifsp.edu.br

Luciano da Silva Carvalho

Graduando em Engenharia Eletrônica
Instituto Federal de São Paulo, São Paulo, SP
luciano.carvalho@aluno.ifsp.edu.br

Carlos Antônio de Souza Galdino Junior

Licenciando em Física
Instituto Federal de São Paulo, São Paulo, SP
carlos.galdino@aluno.ifsp.edu.br

José Vinicius Gonçalves

Licenciando em Física
Instituto Federal de São Paulo, São Paulo, SP
jose.vinicius@aluno.ifsp.edu.br

Maria Nicole Nascimento Silva

Bacharelada em Engenharia Civil
Instituto Federal de São Paulo, São Paulo, SP
maria.nicole@aluno.ifsp.edu.br

RESUMO

Simuladores digitais vêm sendo estudados como ferramentas auxiliares na educação, sobretudo em aulas de física, com o objetivo de engajar o interesse dos discentes e facilitar a compreensão do conteúdo. Nesse sentido, o presente artigo investiga o uso da *Godot Engine*, ferramenta utilizada no âmbito do projeto Física Visual do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (campus São Paulo), para o desenvolvimento de simuladores de física gamificados; processo que se mostrou desafiador, uma vez que a equipe precisou ser capacitada no uso do *software*. Pretende-se, assim, analisar as problemáticas envolvendo a adaptação de fórmulas da física clássica para utilização dentro da engine, além de apresentar as soluções encontradas para que o simulador pudesse atender às condições de gamificação e de simulação da realidade.

Palavras-chave: Simuladores. Educação. Física. Godot.

1 INTRODUÇÃO

Os simuladores podem ser definidos como modelos artificiais que permitem a reprodução parcial ou total de tarefas, podendo ser utilizados em diversos âmbitos, sobretudo no ambiente educacional, como são exemplos suas aplicações na medicina e na aviação (FILHO e SCARPELIN, 2007). É no ensino básico, entretanto, que passaram a ser utilizados com mais frequência, principalmente durante o período pandêmico (DA SILVA, et.al., 2021). Para Ribeiro (2024), por exemplo, a integração da bastante conhecida plataforma Phet no ensino médio tem surtido efeitos bastante positivos se comparada às aulas expositivas, uma vez que garantiu maior participação, melhoria no desempenho e, também, incentivo ao aprendizado dos discentes.

Isso ocorre porque, como apontado por Lopes, Souza e Gomes (2023), os simuladores têm a capacidade de estimular o interesse dos discentes acerca do conteúdo, o que os permite explorar conceitos complexos de maneira mais interativa e envolvente. Dessa forma, eles têm se mostrado ferramentas auxiliares no ensino de ciências como a Física, que é frequentemente vista como difícil pelos estudantes (MOREIRA, 2021).

Para a criação desses simuladores ou simulações, são necessárias ferramentas que permitam esse desenvolvimento. Uma possibilidade é o Geogebra, software de matemática que permite, através de seus recursos, a elaboração de simulações que envolvam conceitos matemáticos (BARBOSA, 2013), bem como simuladores computacionais para o ensino de ciências naturais (ARAUJO, BRACHO, 2020)

Nesse sentido, é explorada a utilização dos motores de desenvolvimento de jogos (games engines) para a confecção de simuladores computacionais gamificados, uma vez que esses motores são ferramentas que permitem a realização de tarefas comuns relacionadas a jogos, possuindo mecanismos que facilitam essas criações, o que torna possível a criação de simulações do mundo real (Paul, Goon, Bhattacharya; 2012). Dessa forma, por se tratar de uma game engine, é evidente que o uso da plataforma possibilita a criação de simuladores com aspectos de jogos, como um sistema de pontuação. Assim, proporciona-se aos usuários uma experiência educativa dos fenômenos físicos ao mesmo tempo que se garante a interatividade inerente à gamificação.

Alguns exemplos de games engines incluem: Unity, Unreal, Game Maker e etc., que possuem simuladores bem conhecidos, como Universe Sandbox, Microsoft Flight Simulator, Train Sim World, Assetto Corsa e entre outros. Entre as diversas opções disponíveis, o Godot aparece como uma ferramenta de código aberto (open source), podendo ser executada em diversos sistemas operacionais, como Windows e Linux, e permitindo a criação de jogos em plataformas desde dispositivos móveis à computadores, além de possuir uma linguagem básica de programação (GDScript), que se caracteriza pela clareza dos códigos, útil à iniciantes na área de programação (SILVA, YEPES; 2018). No entanto,

apesar de suas características, suas produções são muito pouco difundidas e os possíveis simuladores produzidos no software praticamente desconhecidos até o momento.

Diante de uma escassa produção na literatura em português sobre o assunto, vê-se a necessidade de um aprofundamento de caráter teórico e prático da utilização do Godot para o desenvolvimento de simuladores digitais, visando o auxílio do ensino de física. Assim, com o intuito de oferecer uma contribuição para o tema, o presente texto traz uma análise do software a partir da confecção de diversos protótipos.

2 OBJETIVOS

Este artigo visa relatar a prática do desenvolvimento de simuladores gamificados para ensino de física através da Godot Engine 3.5.5, ressaltando dificuldades e acertos para auxiliar a confecção de outros simuladores. Além disso, pretende-se, também, ampliar a literatura disponível acerca do tema.

3 METODOLOGIA

Foi escolhida, a priori, a Godot Engine por indicação do orientador do projeto. Um dos motivos para isso foi a falta de infraestrutura do IFSP - SPO no que tange a defasagem dos computadores existentes na escola, de tal forma que se viu a necessidade de optar por um software cuja demanda de hardware para processamento fosse menor. Outrossim, por se tratar de um aplicativo open source - isto é, de código aberto para usuários -, não há impeditivos quanto ao licenciamento dos jogos ou simuladores nele programados.

Assim, em um primeiro momento, seguiu-se com a etapa de capacitação da equipe, onde foram escolhidos alguns cursos da Udemey (escola online de cursos preparatórios em tecnologia), sendo esses: “Como criar games de plataforma 2D com a Godot Engine”, cuja duração era de 36 horas; e “Domine o Desenvolvimento de Jogos 3D na Godot 4.0 (2023)”, com uma duração de 13 horas. Nesse sentido, a capacitação permitiu introduzir e familiarizar os membros da equipe com a linguagem de programação utilizada pela engine (GDScript) e com sua interface. Dessa forma, as mecânicas introduzidas nos protótipos desenvolvidos basearam-se apenas nas habilidades adquiridas nos cursos realizados - envolvendo, por exemplo, sistemas de pontuação, inserção de inimigos, programação de pontos de vida para o personagem e afins -, além da própria experiência obtida no processo.

Na sequência, partiu-se para o desenvolvimento de protótipos de diversas temáticas, totalizando 4 simuladores iniciais, que poderiam ser feitos com base em projetos já criados e de código aberto disponíveis dentro da própria plataforma, abordando temas voltados à cinemática. Após isso, partiu-se de uma demanda, no âmbito do projeto Física Visual, para a confecção de 3 simuladores gamificados que atendessem o contexto de uma turma específica.

4 RESULTADOS E DISCUSSÃO

Inicialmente, foram criados 4 protótipos de maneira simultânea. Um deles foi o "Jogo do Bloco" (<https://jos3v1n1.itch.io/jogo-do-bloco>¹), no qual o usuário controla o personagem “*player*” que empurra um bloco em repouso, fazendo-o percorrer uma distância conforme o momento linear do jogador. Para isso, foram editados dois nós do tipo *kinematic body*, que permitem a criação de corpos móveis via programação; além de um cenário criado usando *tilemaps* - nó destinado a armazenar e posicionar objetos para construção de cenários. Também se configurou a gravidade do personagem e do bloco com uma constante chamada "GRAVIDADE" e mapeou-se o movimento do personagem pelas teclas "A" e "D". Assim, as regiões de colisão dos dois corpos foram criadas com nós "*CollisionShape*", permitindo a interação entre eles e a superfície, sendo que a área de colisão da superfície foi definida através do *tilemap*.

Outro protótipo, que ainda não possui título, baseou-se em um jogo de plataforma 2D (<https://jos3v1n1.itch.io/2d-plataformer>) com inimigos com uma programação similar ao anterior. Nele, implementou-se duas réguas e criou-se um relógio - através de um nó chamado “*timer*” -, a fim de possibilitar o cálculo de velocidade dos personagens e de projéteis.

Com os dois projetos anteriores, evidencia-se que aplicar as leis da física de movimento na *game engine* utilizada é uma tarefa complexa, pois, muitas vezes, códigos idênticos podem apresentar comportamentos distintos em diferentes projetos, o que dificulta a programação. Isso levou à criação de diversas versões para cada simulador, além do abandono temporário do segundo protótipo. Ademais, outro grande obstáculo foi a conversão de unidades reais para a linguagem de programação, que seria essencial para a criação de simuladores condizentes à realidade.

Nomeado como “Maquinando Trens” (<https://jos3v1n1.itch.io/maquinando-trens>), esse protótipo utilizou como base um jogo de trens, no qual o usuário pode controlar a velocidade de uma locomotiva com vagões em 6 diferentes fases, sob uma perspectiva isométrica. É possível também verificar grandezas referentes ao trem, como sua massa e velocidade, além da força de atrito entre o trilho e o trem; e a resistência do ar. Nesse simulador, foram confeccionados o cenário e a textura do trem e programadas as variáveis para serem exibidas na tela.

Apesar de finalizado, o protótipo apresentou dificuldades de compreensão dos mecanismos utilizados pelo autor do projeto, especialmente no que se refere à interação entre os trilhos e o trem, visto que, em vez de um “*kinematic body*” ou outro nó de corpo, o autor utilizou ferramentas mais complexas para limitar o movimento do trem sobre trilho.

¹ Esse e os outros estão na plataforma *itch.io* - uma plataforma gratuita de upload de jogos - e não necessitam de login. Recomenda-se utilizar o navegador Google Chrome pelo computador para acesso.

Por fim, intitulado “Jogo do Canhão”(<https://jos3v1n1.itch.io/jogo-do-canhao>), o último protótipo desse processo meio momento dispunha, inicialmente, de um canhão e de alguns alvos. Entretanto, pelos mesmos motivos do protótipo “Maquinando trens”, foi realizada uma segunda versão do mesmo, que não contou com a utilização de um projeto *open source*. Dessa forma, a nova versão dispõe de um canhão móvel e alvos fixos em diferentes alturas, cabendo ao usuário configurar as variáveis massa do canhão, massa do projétil lançado, gravidade e ângulo de saída do projétil. Apesar disso, ambos trabalham o conceito de lançamento oblíquo, buscando simular as grandezas nele envolvidas.

Assim, com os dois últimos projetos ficou evidente que editar projetos pré-existent é uma tarefa que, apesar de adiantar a programação de procedimentos básicos, dificulta o desenvolvimento dos simuladores, dada a necessidade de compreender a lógica do autor original. Portanto, para os projetos seguintes, optou-se por confeccioná-los autoralmente.

Dessa forma, a segunda etapa - que partiu de uma demanda -, consistiu na realização de 3 simuladores, também ente, sendo eles: O protótipo “Lâminas Bimetálicas” (<https://jos3v1n1.itch.io/laminas-bimetalicas>) foi o simulador de maior desempenho, possuindo uma gaveta que contém 4 diferentes materiais que podem ser escolhidos pelo clique do usuário para compor a lâmina, que em seguida deve ser direcionada para um espaço em um circuito elétrico, onde ela sofre dilatação. Esses mecanismos foram criados através de nós do tipo “*Button*”, que ao serem clicados emitem sinais e exibem o material referente ao botão como parte da lâmina. Nesse sentido, a vitória é alcançada quando a lâmina com maior deformação é escolhida.

Durante o processo de confecção desse simulador, percebeu-se que a ferramenta de mapeamento do mouse utilizada para arrastar a lâmina apresentou inconsistências no rastreamento do preciso do movimento, erro que é ainda mais visível em outras proporções de tela. Devido a isso, vê-se a necessidade, em versões futuras, de abandonar essa mecânica e utilizar uma mais simplificada.

Agora com “Dilatação linear em trilhos” (<https://jos3v1n1.itch.io/trem-dilatacao>), foram programadas, nesse simulador, as relações entre temperatura, tempo, coeficiente de dilatação do trilho, comprimento do segmento de trilho, comprimento do percurso e o espaçamento entre trilhos - método antigamente usado para evitar a dilatação. Enquanto as 4 últimas variáveis devem ser determinadas pelo usuário, a temperatura varia de acordo com valores compilados de 2014 a 2024 (dados referentes ao Rio de Janeiro), através de “arrays” e “dicionários”, que permitem o armazenamento de diversos valores e seu acesso de maneira programada.

Nomeado Jogo do elétron (<https://jos3v1n1.itch.io/jogo-do-eletron>), esse teve como foco a exploração da corrente elétrica com base no Modelo de Drude. Para tal, buscou-se confeccionar átomos referentes ao material do fio e elétrons livres, de tal forma que deveria ocorrer uma interação entre eles. Entretanto, viu-se dificuldade em programar essas colisões, uma vez que o elétron,

segundo o modelo, deve ser repellido de maneira coerente ao ângulo de colisão, obstáculo que ainda não foi solucionado.

Como alternativa, foi proposto um simulador sobre calorimetria na cozinha, intitulado “Física na Cozinha” (<https://jos3v1n1.itch.io/simulador-estados-da-materia>), focando, também, os estados da matéria. Esse protótipo foi criado de forma semelhante à gaveta do simulador “Lâminas bimetálicas”, e permite que o usuário possa selecionar itens de cozinha, como microondas e fogão, para fundir o gelo, dada a potência dos aparelhos - a partir da qual se determina o tempo necessário para a fusão ocorrer.

Assim, apesar de diversos problemas atribuídos ao Godot, entende-se que a plataforma dispõe, de fato, de um grande potencial, à medida em que permitiu a confecção dos simuladores integralmente em seu software, além de um dos protótipos sob demanda já ter sido finalizado. Outrossim, destaca-se que o desempenho da *engine* se manteve constante mesmo com projetos maiores e com maior demanda de hardware.

5 CONSIDERAÇÕES FINAIS/CONCLUSÃO

A criação desse e dos outros simuladores por meio da Godot Engine envolveu uma série de desafios. A princípio, uma das críticas surge em torno do uso da linguagem própria da Engine, o GDScript, cuja sintaxe carece de algumas construções mais simples, como ocorre em outras linguagens, tais como funções de *delay* - para controle de tempo de ações - e de randomização.

Esses desafios podem ser, em parte, atribuídos à etapa de capacitação, uma vez que os cursos utilizados focaram em demasiado no desenvolvimento de jogos, em vez de simuladores, que exigem a adaptação da física do mundo real para o mundo digital. Como um indicativo, os comandos executados pelos professores dos cursos já apresentavam erros quando não se comportavam da mesma maneira ao serem transcritos, o que atrasou o andamento dos cursos.

Assim, notou-se que para o grupo desenvolver os simuladores, foi necessário um aprendizado constante e árduo, pautado na tentativa e erro, mostrando que os cursos escolhidos não bastaram para o objetivo. Sendo assim, segue-se que o mais adequado seriam cursos focados na aprendizagem de programação em GDScript e, se possível, em simuladores de física. No entanto, apesar das dificuldades, foi possível criar protótipos de considerável desempenho. Portanto, acredita-se que com uma maior experiência com a plataforma, o *Godot* pode ser uma game engine interessante para o desenvolvimento de simuladores de destaque para o cenário educativo.

REFERÊNCIAS

BARBOSA, Sandra Malta. O software Geogebra e as possibilidades do trabalho com animação. Revista do Instituto GeoGebra Internacional de São Paulo. Disponível em: <https://revistas.pucsp.br/IGISP/article/view/12843/12201>. Acesso em: 29 jul. 2024.

DA SILVA, Ana Suellen Gomes; DEOSTI, Leonardo; DE CARVALHO, Hercília Alves Pereira; CARVALHO, Hercília Alves Pereira de. Educação em tempos de pandemia: uma experiência no ensino de física. UFPR, 2021. Disponível em: <https://compartilha.ufpr.br/wp-content/uploads/2021/11/9-educacao-em-tempos-de-pandemia-1.pdf>. Acesso em: 20 jul. 2024.

DA SILVA, Vitor Poltronieri; YEPES, Igor. Desenvolvimento de jogos na plataforma Godot. Disponível em: <http://anais.eati.info:8080/index.php/2019/article/view/282/294>. Acesso em: 29 jul. 2024.

GUTIÉRREZ-ARAUJO, Rafael Enrique; CASTILLO-BRACHO, Luis Andrés. Simuladores com o software GeoGebra como objetos de aprendizagem para o ensino da física. Disponível em: http://www.scielo.org.co/scielo.php?pid=S0121-38142020000100201&script=sci_arttext. Acesso em: 29 jul. 2024.

LOPES, J. S.; SILVA, A. G. da S.; DE SOUZA, G. F. de S. Ensino de Física com uso de simuladores virtuais: potencial de utilização em sala de aula. Disponível em: <https://www2.ifrn.edu.br/ojs/index.php/HOLOS/article/view/14365>. Acesso em: 26 jul. 2024.

MOREIRA, Marco Antonio. Desafios no ensino da física. Revista Brasileira de Ensino de Física, 30 jun. 2021. Disponível em: <https://www.scielo.br/j/rbef/a/xpwKp5WfMJsfCRNFCxHqLy/>. Acesso em: 29 jul. 2024.

PAUL, Partha Sarathi; GOON, Surajit; BHATTACHARYA, Abhishek. History and comparative study of modern game engines. Disponível em: <https://encurtador.com.br/NMNNp>. Acesso em: 29 jul. 2024.

PAZIN FILHO, Antonio; SCARPELINI, Sandro. Simulação: definição. Medicina (Ribeirão Preto), v. 40, n. 2, p. 162-166, 2007. Disponível em: <http://www.revistas.usp.br/rmrp/article/view/312>. Acesso em: 29 jul. 2024.

RIBEIRO, João Pedro Mardegan. A integração do laboratório virtual “PHET Interactive Simulations” no ensino de física. Disponível em: <https://ciet.ufscar.br/submissao/index.php/ciet/article/view/797>. Acesso em: 26 jul. 2024.