

# O jogo educacional Codeland como instrumento de apoio ao ensino de lógica de programação com Python

A Silva, G Nascimento, J Oliveira, L Neto, M Araújo, V Oliveira,

**Abstract**—As students are being introduced to programming disciplines, some might have difficulties adapting to the teachers' methodology. Very often, this stems from the fact that the teaching process used doesn't suit all of the students. Thus, the purpose of this article is to show Codeland, a serious game under development with the objective of supporting logic and programming education with Python especially in higher education courses. Therefore, the game brings aspects like the ease of being applied in introductory programming disciplines and having exercises with the proposal of improving the thinking capacity of the students, being considered by the majority who tested it as an appropriate learning tool. Based on the ENgAGED development model, an initial version was developed with the MEEGA+ evaluation model being adopted so that students of introductory programming courses could validate it.

**Index Terms**—Serious Games, Programming Introduction, Graduation, University Education

## I. INTRODUÇÃO

O emprego de jogos educacionais como ferramenta de apoio no ensino da programação tem gerado resultados comprovadamente positivos [1]. De acordo com [2] os jogos têm contribuído significativamente não só em disciplinas de programação, mas nas disciplinas de computação como um todo, promovendo um maior engajamento dos alunos. A disciplina de programação é considerada uma das disciplinas mais difíceis e impactantes para alunos ingressantes em cursos de computação [3]. Isto posto, denota-se que jogos educacionais com uma boa proposta de aprendizagem são altamente viáveis como ferramenta de apoio ao ensino de programação. Métodos tradicionais não são suficientes para cativar os alunos na disciplina de programação e acabam criando um ambiente entediante e desmotivante [4]. O resultado é a falta de interesse pela disciplina, que tem por sua vez, provocado altos índices de reprovação e evasão [5]. Desse modo, a aplicação de meios que promovam a abordagem do conteúdo de maneira dinâmica pode atenuar esses índices, por conseguinte Codeland traz essa proposta. Sendo assim, o objetivo do artigo é apresentar o Jogo

Codeland, um jogo educacional que se encaixa no gênero Role Playing Game (RPG) e tem como finalidade auxiliar o aluno no processo de aprendizagem de programação Python em disciplinas de programação de cursos ligados a computação. Esse trabalho é constituído de maneira que, na Sessão 2 é apontado algumas características de trabalhos relacionados e área em aberto. Por conseguinte é apresentada a metodologia de desenvolvimento na Seção 3. Codeland é dissertado na Seção 4 e finalmente são discutidas as considerações finais e planejamentos para trabalhos futuros na Seção 5.

## II. TRABALHOS RELACIONADOS

No trabalho de [6] foi criado um MMORPG (Massive Multiplayer Online Role-Playing Game – Jogo de Interpretação de Personagem Online e em Massa para Múltiplos Jogadores) onde o professor (que pode ser de qualquer área ou disciplina) tem a oportunidade de introduzir tarefas para que os alunos respondam dentro do próprio jogo. Os autores enfatizaram que MMORPGs educacionais são divertidos para jogar e fornecem boas experiências no processo de aprendizagem do aluno. Todavia, apesar da proposta ser interessante, o jogo enfatiza questões teóricas, o que pode ser insuficiente para um bom processo de aprendizagem em disciplinas de programação. Já [7] produziram um jogo no qual um robô é encarregado de recolher discos falhos e realizar correções por meio de linhas de código com objetivo de ajustar a sua nave espacial. O jogo é denominado sCool e o público alvo são jovens. Villareale et al [8] consideram a reflexão como um importante aspecto na aprendizagem de programação. Essa ideia equivale ao ciclo de pensar, fazer e observar, sendo uma prática importante em contextos nos quais existem múltiplas soluções. Dessa maneira, pode ser incorporada em um jogo por meio da execução em passo-a-passo da solução do aluno, perguntas esporádicas sobre os assuntos abordados no jogo, soluções exemplos e/ou interação com demais jogadores. Com base na literatura, e principalmente nos resultados apresentados nessa seção, é possível concluir que a aplicação de jogos no ensino se trata de algo factível, sobretudo quando associados à programação.

## III. METODOLOGIA DE DESENVOLVIMENTO

O processo de desenvolvimento ENgAGED (Educational Games Development), desenvolvido por [9], foi utilizado pelos pesquisadores deste projeto. Assim é possível desenvolver, de forma sistemática, jogos sérios com foco em aprendizagem

Ayrton Lucas Silva, Instituto Federal da Paraíba, ayrtton.lucas@academico.ifpb.edu.br

Gabriel da S. Nascimento, Instituto Federal da Paraíba, nascimento.gabriel@academico.ifpb.edu.br

Jhonatan Guilherme Oliveira, Instituto Federal da Paraíba, jhonatan.oliveira@academico.ifpb.edu.br

Luiz Medeiros Neto, Instituto Federal da Paraíba, luiz.neto@academico.ifpb.edu.br

Maria Eduarda Cunha Silva Araújo, Instituto Federal da Paraíba, eduarda.cunha@academico.ifpb.edu.br

Victor Andre Pinho de Oliveira, Instituto Federal da Paraíba, victor.oliveira@ifpb.edu.br

eficiente a partir de 5 fases: Análise, Projeto, Desenvolvimento, Execução e Avaliação. Nesse sentido, é possível identificar a etapa de desenvolvimento como sendo uma das principais. Assim, é factível citar que algumas seções do desenvolvimento como a análise de requisitos citada anteriormente; a etapa de concepção, que define as informações do jogo, itens, tarefas e afins; o processo de design e implementação, que refere-se a criação dos elementos visuais possibilitando a execução no jogo dos itens definidos na concepção e dos elementos de design. Por conseguinte, é executada a etapa de testes que valida a implementação. Convém lembrar, ainda, sobre as etapas de Execução e Avaliação do projeto. Na fase de execução, o jogo é posto em funcionamento e caso seja encontrada alguma falha, os desenvolvedores irão corrigi-la antes de caminhar para a próxima etapa. Na avaliação, são realizados os testes com os alunos notificando as adversidades encontradas para correções futuras. Visando o auxílio no desenvolvimento dos itens relacionados ao jogo, foram utilizados softwares com funções específicas, sendo eles o Tiled e o RPGMaker MV, o que permitiu um desenvolvimento rápido e robusto do jogo. A figura 1 (a), diz respeito ao Tiled, um editor de cenários, em que são empregados itens visuais para gerar cenários 2D. Por sua vez, o RPG Maker MV, representado na figura 1 (b), é uma engine para a criação de jogos 2D, que visa principalmente o estilo RPG. Essa ferramenta, acelera o processo de produção, visto que possui uma codificação por eventos e a adição de código individuais através de um plugin, gerando uma gama de resultados no jogo que são apresentados posteriormente.

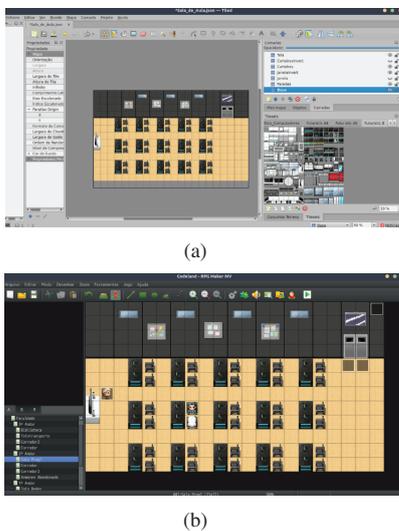


Fig. 1. Desenvolvimento de cenário: (a) No Tiled, (b) No RPG Maker MV

A aplicabilidade da linguagem Python diretamente no RPG Maker não é possível, visto que o mesmo reconhece apenas JavaScript. Logo, foi desenvolvido um plugin (visualizar a figura 2 a) para a interpretação da linguagem Python, elaborado de maneira que permitisse o exercício da reflexão por parte dos jogadores, uma vez que essas são algumas das motivações para desenvolvimento da ferramenta. Portanto, com o script estruturado dentro do jogo ele permite que o jogador reflita sobre um determinado problema e logo depois de visualizá-lo, realize a escrita de linhas de código que possibilitem a

correção da adversidade proposta (visualizar a figura 2 b). Após esse processo, o plugin irá realizar uma análise em cima do que o usuário informou, respaldando-se numa gramática que torna a análise abrangente. Com isso, após verificar a linha escrita é apresentado na tela uma mensagem informando acerto ou erro com base na questão que foi proposta pelo desafio.

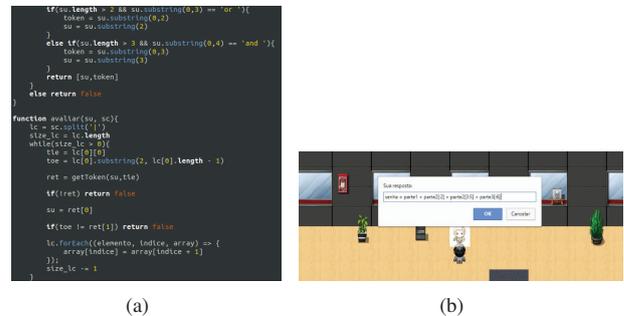


Fig. 2. Plugin para interpretação do Python: (a) Código, (b) Evento executado no jogo.

## IV. CODELAND

### A. Objetivos

O jogo está sendo desenvolvido visando contemplar o conteúdo programático das disciplinas introdutórias de programação, que são: (1) Raciocínio lógico, (2) Tipos de dados Python, (3) Operadores e expressões, (4) Entrada e Saída, (5) Estruturas de Controle: Sequencial, Condicional e de Repetição, (6) Funções e recursividade.

### B. O Processo de aprendizagem com o jogo

Dentre os tópicos citados, até a escrita do artigo, a versão beta do jogo aborda os tópicos (1), (2) e (3), todos inseridos cuidadosamente ao contexto e à narrativa do jogo. O tópico (1) por meio de desafios no estilo puzzle dentro do jogo, como exemplo o puzzle “liga fios” onde o jogador deve ligar fios cortados criando um caminho válido para que a energia passe e ele obtenha acesso à rede de comunicação. E os tópicos (2) e (3), por meio de pequenas tarefas de programação. Nelas são apresentados códigos adulterados ao jogador, nos quais apenas uma das linhas está errada - no contexto do jogo é abordado por objetos de Codeland que tiveram sua programação alterada pela IA - e sua missão é inserir na caixa de texto a correção da linha de código que restaure o funcionamento do objeto. É uma fase que impede o jogador de progredir caso o código não esteja correto mas permite que ele tente quantas vezes forem necessárias. A título de exemplo temos a tarefa “porta com senha”. Nela o jogador se depara com uma porta que não abre, pois a IA alterou o código que processa a senha de acesso. Para solucionar o problema, o jogador tem de lidar com slicing de string e concatenação.

## V. RESULTADOS E AVALIAÇÕES

Toda a pesquisa foi realizada de forma remota em razão da pandemia da COVID-19 e logo após a fase de Desenvolvimento da versão inicial de Codeland, foram iniciadas as fases

de Execução e Avaliação do jogo. Participaram dessas fases 12 alunos voluntários matriculados nos cursos de Engenharia da Computação e Telemática que estavam cursando a disciplina de Introdução à Programação. Os alunos voluntários receberam instruções precisas de como baixar, instalar e executar o jogo. Após a experiência de jogar Codeland, os alunos foram submetidos ao questionário MEEGA+, um modelo sistematicamente desenvolvido para avaliar a qualidade dos jogos educacionais [10]. O modelo divide-se essencialmente em duas partes, uma que coleta dados demográficos e outra que coleta dados sobre usabilidade e efetividade, isto é, se o jogo de fato contribuiu para o aprendizado. Frisa-se que a avaliação realizada tem ênfase na experiência do jogador e na sua percepção da aprendizagem. Dos alunos participantes, 9 eram do sexo Masculino e apenas 3 do sexo Feminino; 10 estavam entre 18 e 28 anos, 1 abaixo de 18 e 1 entre 29 e 39. Em relação à frequência do participante em jogar jogos digitais, 7 afirmaram que jogam semanalmente, 2 que jogam diariamente, 2 que jogam raramente e 1 que joga mensalmente. As avaliações sobre a experiência do jogador podem ser vistas na Figura 3 e possuem o maior peso na fase de Avaliação, pois determinam a efetividade e relevância do jogo em relação ao conteúdo abordado. No geral, pode-se observar um alto nível de concordância (+1 ou +2). Também foram observadas muitas respostas indiferentes (0) - geradas, provavelmente, pela cobrança dos autores no retorno da avaliação em decorrência do avanço do tempo -, mas que não podem ser consideradas como respostas negativas.

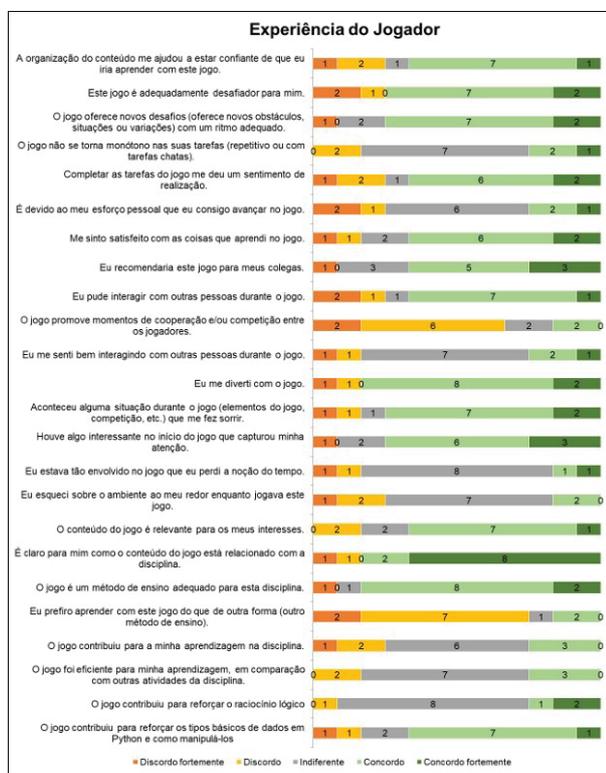


Fig. 3. Avaliação da experiência dos jogadores

Um aspecto que se destacou no âmbito da percepção da

aprendizagem foi o de que “O jogo contribuiu para reforçar os tipos básicos de Python e como manipulá-los”, tendo 66% de avaliações positivas. Acredita-se que esse resultado foi alcançado por ter sido bem explorado nas tarefas de programação. Ademais, 83% consideraram o jogo como ferramenta adequada para a disciplina, muito embora 83% não tenham preferência para aprender com Codeland. Julga-se que o resultado dessas avaliações são uma consequência do jogo ainda estar na sua versão beta, fazendo com que alguns aspectos que melhoram a imersão dos alunos no jogo, (como por exemplo, a história ser passada de uma forma rápida e ele não possuir gratificações para os jogadores quando resolvem desafios) ainda não estejam totalmente desenvolvidos.

## VI. CONSIDERAÇÕES FINAIS

Codeland promove uma adição aos métodos utilizados para ensino de lógica e programação nos cursos relacionados à área de computação, como mostrado neste artigo. A versão beta do jogo proporcionou experiências positivas nos jogadores, sendo desejável que uma diferente imersão nos assuntos tratados em aula tanto pelo professor, quanto pelos alunos seja possível. A proposta do jogo se faz bastante aconselhável quando se trata de uma nova abordagem descontraída da programação, adaptando o ensino às novas gerações criadas em contato direto com telas de computadores e celulares. Em trabalhos futuros serão também consideradas às críticas recebidas pelos alunos, como a história ser muito rápida, ter poucas tarefas de programação e a correção de outros bugs. Além disso, é pretendido que novos tópicos sejam revisados e adicionados para incrementar a aplicabilidade do jogo em demais conteúdos da disciplina de programação.

## REFERÊNCIAS

- [1] Shahid, M. et al. (2019) “A Review of Gamification for Learning Programming Fundamental”. In: International Conference on Innovative Computing, n. 3, Laore. Proceedings, p.1-8.
- [2] HOSSEINI, H.; HARTT, M.; MOSTAFAPOUR, M.. Learning IS Child’s Play: Game-Based Learning in Computer Science Education. ACM Transactions on Computer Education, n. 19, 18 p., Junho 2019.
- [3] Caspersen, M. E. e Kölling M. (2009) “STREAM: a first programming process”. In: ACM Transactions on Computer Education, n 9, pp. 29.
- [4] BORNA, Keivan; RAD, H. Maghami. Serious Games in Computer Science Learning Goals. In: DIGITAL GAMES RESEARCH CONFERENCE: TRENDS, TECHNOLOGIES, AND APPLICATIONS, no 2, 2018, Teerão. Proceedings. [S.I.]:IEEE, 2019, p. 161-166.
- [5] Monclar, R. S. et al. (2018) “Jogos com Propósito para o Ensino de Programação”. In: Simpósio Brasileiro de Jogos e Entretenimento Digital, n. 17. Foz do Iguaçu. Proceedings of SBGames, p. 1132-1140.
- [6] WANG, A. I.; WU, B.. An Application of Game Development Framework in Higher Education. International Journal of Computer Games Technology, no 2009, p. (6):12, 2009.
- [7] Steinmaurer, A. et al. (2019) “sCool - Game-Based Learning in Computer Science Class”. International Journal of Engineering Pedagogy, n. 9, p. 35-50.
- [8] VILLAREALE, Jennifer et al.. Reflection in Game-Based Learning: A Survey of Programming Games. In: INTERNATIONAL CONFERENCE ON THE FOUNDATIONS OF DIGITAL GAMES, no 20, 2020, Malta. Proceedings. New York: Association for Computing Machinery, 2020, p. 1-9.
- [9] Battistella, P. E., e von Wangenheim, C. G. (2016) “ENgAGED: Um processo de desenvolvimento de jogos para ensinar computação”. In: Congresso Brasileiro de Informática na Educação, v. 27, no 1, p. 380-389.

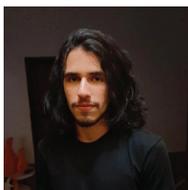
- [10] Petri, G., von Wangenheim et al. (2019) "MEEGA+: Um Modelo para a Avaliação de Jogos Educacionais para o ensino de Computação". In: Revista Brasileira de Informática na Educação. Brazilian Journal of Computers in Education, v. 27, n. 3.



**Ayrton L. Silva** Ayrton L. Silva Graduando em Engenharia de Computação pelo Instituto Federal de Educação, Ciências e Tecnologia da Paraíba. Interesse em pesquisas nas áreas de inteligência artificial, jogos sérios, desenvolvimento de software.



**Gabriel da S. Nascimento** Graduando em Engenharia de Computação pelo Instituto Federal de Educação, Ciências e Tecnologia da Paraíba. Possui ensino técnico em Telecomunicações pela Escola Técnica Redentorista, além de conhecimentos na área de programação (Python, C/C++). Atua na área de desenvolvimento de software de otimização com python 3. Interesse em pesquisas nas áreas de inteligência artificial, sistemas embarcados e desenvolvimento de software.



**Jhonatan G. Oliveira** Jhonatan Guilherme O. Pereira. Graduando em Engenharia de Computação pelo Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Campina Grande. Possui conhecimento em linguagens de programação como Python, C/C++ e PHP. Atua na área de desenvolvimento de software de otimização com Python3. Interessado nas áreas de ciência de dados e desenvolvimento de software.



**Luiz M. Neto** Luiz Medeiros Neto. Graduando em Engenharia de Computação pelo Instituto Federal de Educação, Ciências e Tecnologia da Paraíba. Dispõe de conhecimentos em sistemas operacionais Linux e Microcontroladores. Dentre as linguagens de programação nas quais possui afinidade estão enumeradas Python, C/C++ e Java. Além disso, durante a graduação também foram adquiridos conhecimentos em MySQL, fundamentos da Web e Node.js. Dessa maneira, detém interesse nas áreas de Sistemas Embarcados, Desenvolvimento e Engenharia de Software.

nharia de Software.



**Maria Eduarda C. S. Araújo** Maria Eduarda C. S. Araújo Graduanda em Engenharia de Computação pelo Instituto Federal de Educação, Ciências e Tecnologia da Paraíba. Atua na área de testes de software. Interesse em pesquisas nas áreas de inteligência artificial, realidade virtual e sistemas embarcados.



**Victor Andre P. de Oliveira** Mestre em Ciência da Computação. Victor A. P. Oliveira atua como professor do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB desde 2012. Leciona principalmente disciplinas relacionadas a área de Programação, como: Introdução à Programação de Computadores, Programação Orientada a Objetos, Laboratório de Programação, Estruturas de Dados, mas não se limitando a elas. No âmbito da pesquisa, possui interesses nas áreas de Redes Óticas, Game-Based Learning, Segurança Ofensiva e Análise de

Aprendizagem. Victor gosta de aprender e de se aventurar por novas áreas do conhecimento e está sempre disposto a compartilhar de uma forma didática e empolgante o que tem aprendido.