

Codeland: um jogo para o ensino de lógica de programação com Python

Ayrton L. Silva, Gabriel S. Nascimento, Jhonatan G. O. Pereira, Luiz M. Neto,
Maria E. C. S. Araújo, Victor A. P de Oliveira¹

¹Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB)
– Campina Grande – PB – Brasil

{ayrton.lucas, nascimento.gabriel, jhonatan.oliveira, luiz.neto, eduarda.cunha}@academico.ifpb.edu.br, victor.oliveira@ifpb.edu.br

Abstract. *The objective of this work is to present Codeland, an educational game in development to assist in the teaching of programming logic with Python. It is a Role-Playing Game being developed in RPG Maker MV following the ENgAGED development process, a model created specifically for the development of educational games. In the evaluation conducted with the students who played the first version of the game, the students' perception of how the game relates to the discipline stood out as a positive aspect, being considered by most as an adequate learning tool.*

Keywords— *Serious Games, Educational Games, Digital Game Based Learning (DGBL), Programming in Higher Education*

Resumo. *O objetivo deste trabalho é apresentar Codeland, um jogo educacional em desenvolvimento para auxiliar no ensino da lógica de programação com Python. Trata-se de um Role-Playing Game que está sendo desenvolvido no RPG Maker MV seguindo o processo de desenvolvimento ENgAGED, um modelo criado especificamente para o desenvolvimento de jogos educacionais. Na avaliação realizada com os estudantes que jogaram primeira versão do jogo, destacou-se como aspecto positivo a percepção dos alunos em como o jogo se relaciona com a disciplina, sendo considerado pela maioria como uma ferramenta adequada de aprendizagem.*

Palavras-chave— *Jogos Sérios, Jogos Educacionais, Aprendizagem Baseada em Jogos Digitais, Programação no Ensino Superior*

1. Introdução

A disciplina de programação é considerada uma das disciplinas mais difíceis e impactantes para alunos ingressantes em cursos de computação [CASPERSEN e KÖLLING 2009]. Por um lado, as dificuldades encontradas, de forma geral, decorrem da falta de habilidades exigidas pela disciplina, como raciocínio lógico, capacidade de

abstração e capacidade de resolver problemas [SHAHID *et al.* 2019]. Por outro lado, tem-se a inadequação da metodologia aplicada pela maioria dos professores, que tende a seguir o modelo instrucionista em detrimento ao modelo construtivista, sendo este último considerado mais adequado para a disciplina [SILVA *et al.* 2018]. O resultado é a falta de interesse pela disciplina, que tem, por sua vez, provocado altos índices de reprovação e evasão [MONCLAR *et al.* 2018].

Nesse contexto, os jogos de computadores voltados para educação – também chamados jogos educacionais –, têm sido utilizados como uma estratégia interessante e bastante eficaz, além de tornar o aprendizado divertido [KADEL *et al.* 2019].

Diante do exposto, o objetivo do artigo é apresentar o jogo Codeland, o qual está sendo desenvolvido segundo o processo ENgAGED [BATTISTELLA e von WANGENHEIM 2016] e teve sua primeira versão *beta* avaliada pelo modelo MEEGA+ [PETRI *et al.* 2019]. Trata-se de um jogo educacional que se enquadra no gênero *Role Playing Game* (RPG) e cuja finalidade é a de auxiliar o aluno no processo de aprendizagem de programação com Python em disciplinas de programação de cursos ligados à Computação.

Esta proposta se justifica pela sua relevância científica e acadêmica. De acordo com Shahid *et al.* (2019), o emprego de jogos educacionais tem gerado muitos resultados positivos. Ademais, há uma tendência para o uso de jogos como parte da metodologia em sala de aula [VOULGARI *et al.* 2020]. Apesar dos últimos avanços no campo da DGBL, ainda há muito espaço para estudo e contribuição, sobretudo em disciplinas de programação [MONCLAR *et al.* 2018], [VAHLICK *et al.* 2020].

O presente artigo se encontra organizado como descrito a seguir. Na Seção 2 serão apresentados os trabalhos relacionados. A proposta Codeland é apresentada na Seção 3. Por fim, as considerações finais e propostas de trabalhos futuros, na Seção 4.

2. Trabalhos Relacionados: jogos para o ensino da programação

Em uma revisão sistemática realizada previamente, os autores deste trabalho identificaram estudos que relatam propostas relacionadas com o jogo educacional Codeland, as quais são comentadas a seguir.

Em Hartness (2004) é descrito o RoboCode, um jogo de robôs virtuais baseado em Java para o ensino de técnicas de programação da linguagem. O jogo foi desenvolvido pela IBM e consiste basicamente de uma arena virtual e uma interface para o desenvolvimento do robô. Contudo, embora a ideia do jogo seja simples, a estratégia para se vencer o jogo não o é, o que o torna inadequado para disciplinas introdutórias de programação.

Esteves *et al.* (2008) desenvolveram um ambiente virtual em três dimensões (3D) para o ensino da programação. Nele, os estudantes usam avatares (uma representação gráfica do jogador) para criar objetos 3D e programar seu comportamento (usando uma

linguagem baseada em C). Os autores defendem que a visualização é um meio pelo qual são reduzidas as dificuldades, mas perceberam que os alunos tiveram algumas dificuldades em se adaptar ao ambiente tridimensional.

Steinmaurer *et al.* (2019) desenvolveram um jogo educacional para ensino de programação chamado sCool, direcionado a estudantes na faixa etária de 10 a 20 anos. O jogo possui uma narrativa-guia e trata-se de um robô em uma missão de resgate que deve coletar discos defeituosos e completar o código faltante nos discos com o objetivo de fazer sua nave espacial voltar a funcionar.

Vahldick *et al.* (2020) criaram um jogo educacional baseado na programação em blocos para auxiliar no ensino da programação em cursos de computação. O jogo é voltado para iniciantes e, de acordo com os autores, a estratégia de blocos é interessante por permitir que os novos alunos foquem no problema, visto que os blocos representam instruções de mais alto nível.

3. Codeland: a ilha da programação

Nesta seção, é apresentada a proposta do jogo educacional Codeland. Frisa-se que o jogo ainda está em desenvolvimento, mas já possui uma versão *beta* jogável que contempla tópicos iniciais que vão desde o raciocínio lógico até os tipos de dados e operadores em Python. A Figura 1 ilustra a primeira cena, momento em que o jogador é apresentado à narrativa do jogo.

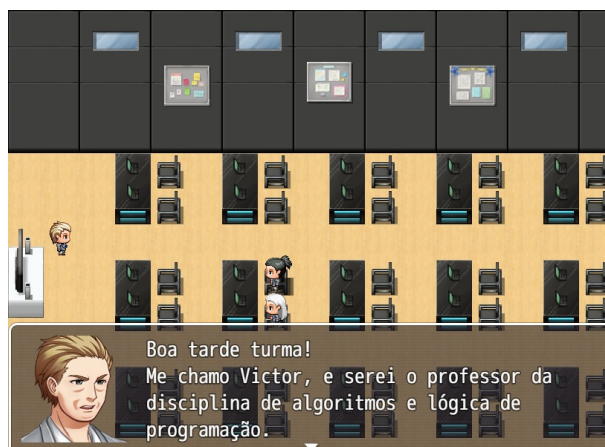


Figura 1. Primeira cena do jogo.

3.1. A proposta

Para que um jogo educacional cumpra o seu papel, qual seja o de servir para construir conhecimento para quem o joga, ele precisa ter um conjunto de características e seguir alguns critérios. Segundo Monclar *et al.* (2018), o desafio, a fantasia e a curiosidade são características que devem estar presentes em bons jogos para o ensino da programação. E de acordo com Rocha (2017), um jogo educacional para ser efetivo deve apresentar,

mas não exaustivamente, os seguintes critérios: objetivos, desafios e informações claras; dificuldade progressiva; facilidade de interação com o jogo; além de promover algum grau de independência ao aprendiz.

Diante do exposto, Codeland está sendo desenvolvido com vistas a atender as demandas encontradas na literatura. Exatamente por essa razão que o gênero escolhido para o jogo foi o RPG, pois, em jogos do gênero, é comum uma narrativa-guia, uma mapa a ser explorado, além de desafios e tarefas para o jogador, colocando-o em um ambiente bastante imersivo [PIMENTEL *et al.* 2019].

3.2. A narrativa

A narrativa de Codeland se passa num futuro distante em uma ilha de mesmo nome onde tudo é programável. Robôs, portas, aspiradores, equipamentos, instrumentos musicais etc., qualquer objeto em Codeland funciona por meio da programação. Na ilha de Codeland, tudo costumava ir muito bem e em perfeita harmonia, até que uma Inteligência Artificial (IA) criada pelo Dr. Guido van Rossum¹ se tornou independente e autônoma de forma inexplicada, rebelando-se contra Codeland a fim de dominá-la.

O que antes era um lugar harmonioso e prazeroso de se viver, tornou-se um pesadelo constante. Agora, Codeland, dominada pela IA, está um caos, pois muitos dos objetos tiveram sua programação alterada e estão apresentando um comportamento estranho e nocivo, afetando a vida de toda a população.

Diante desse cenário caótico, é preciso de um herói – o jogador – que esteja disposto a aprender a programar e cuja missão seja restaurar o que a IA modificou, além de lutar para destruí-la. Enquanto trabalha para cumprir sua missão, o jogador deve enfrentar desafios de lógica e de programação em Python por meio de grupos de tarefas, organizados de forma progressiva, encaixados na narrativa e que atendem a um determinado tópico da disciplina (raciocínio lógico, tipos de dados, operadores etc.).

3.3. Metodologia de desenvolvimento

O desenvolvimento de Codeland tem seguido o modelo ENgAGED (*EducatioNAl GamEs Development*) proposto por Battistella e von Wangenheim (2016), com algumas pequenas adaptações. Trata-se de um processo de desenvolvimento de software criado especificamente para produção de jogos educacionais. Sua finalidade é guiar a criação de jogos de modo que eles cumpram o papel de instruir e se encaixem na metodologia da disciplina. O modelo é organizado em cinco fases: Análise, Projeto, Desenvolvimento, Execução e Avaliação.

Seguindo o modelo, na fase de Análise deve-se definir o que será feito, isto é, deve-se estabelecer o objetivo geral do jogo, a caracterização dos aprendizes e as Unidades Instrucionais (UI) - tópicos da disciplina – que ele vai contemplar. No caso de

1 Uma homenagem ao criador da linguagem Python

Codeland, o objetivo geral do jogo é auxiliar no ensino da lógica de programação com Python. O público-alvo são alunos ingressantes em cursos de Computação, com idades geralmente entre 18 e 28 anos. Até a escrita deste artigo, o jogo contempla as seguintes UIs: raciocínio lógico, tipos de dados Python, operadores e expressões.

Na fase de Projeto deve-se definir como o jogo será feito, o que envolve definir o gênero do jogo e como as UIs serão abordadas ao longo do jogo. No caso de Codeland, o gênero escolhido foi RPG e cada UI organizada em grupos de tarefas com nível progressivo de dificuldade dispostas ao longo da narrativa. Ainda nesta fase, foi definido o RPG Maker MV (ver Figura 2a) como *game engine*, pois é uma ferramenta propriamente voltada ao desenvolvimento de RPGs. Essa versão específica foi escolhida em razão da possibilidade da criação de *plugins* (em javascript) que permitem estender as funcionalidades do jogo. A criação de um *plugin* foi necessária para introduzir um pequeno interpretador de expressões Python ao jogo (ver Figura 2b do código plugin).

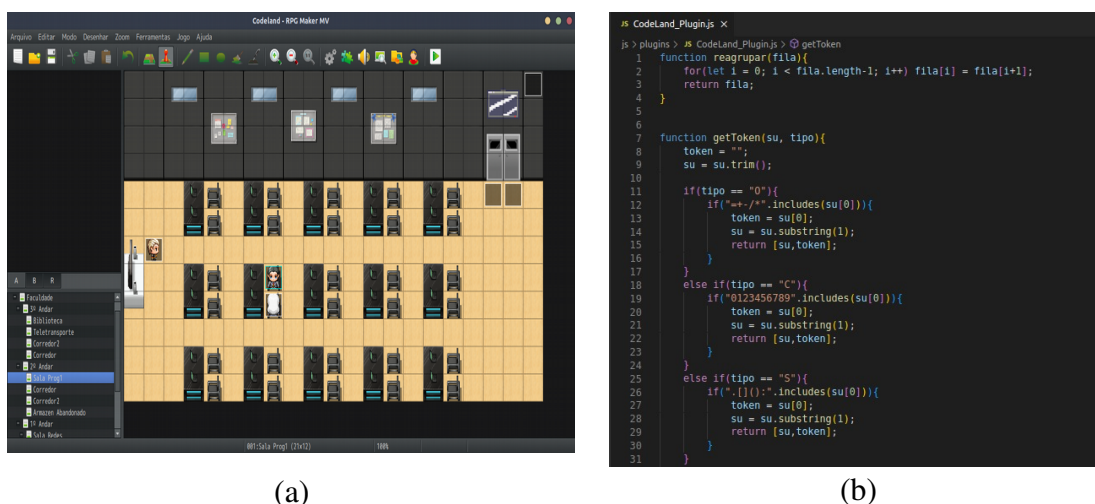


Figura 2. (a) Produção do jogo com RPG Maker MV (b) Trecho do *plugin* desenvolvido em javascript

Continuando, na fase de Desenvolvimento é onde ocorre a criação do jogo propriamente dita. É uma fase dividida em cinco subfases, quais sejam: Análise, Concepção, Design, Implementação e Teste, as quais devem iterar por cada UI definida anteriormente. Convém mencionar que, antes de cada iteração, houve uma extensa discussão entre os pesquisadores deste trabalho a fim de estabelecer a forma sobre como explorar cada tópico na respectiva iteração do jogo. Em relação às subfases, a de Análise deve definir os requisitos e objetivos da iteração, que, na iteração 1 de Codeland, p.e., definiu-se ambientar o jogador, inserindo-o na narrativa e fazendo-o entender sua missão; a de Concepção deve descrever exatamente como a dinâmica deve ocorrer, que, na iteração 1 de Codeland, p.e., ficou estabelecido duas tarefas no formato de *puzzle*; a de Design é a subfase em que o desenho do jogo é feito, o qual, no caso de Codeland, foi no próprio RPG Maker MV; a de Implementação é onde ocorre a programação do jogo,

sendo nessa fase, mais especificamente na segunda iteração, que o *plugin* para interpretar expressões Python foi desenvolvido; por fim, na subfase de Teste, é onde a equipe tem a possibilidade de avaliar se tudo está funcionando adequadamente.

Na fase de Execução ocorre a instalação do jogo e a experimentação por parte da turma. Por fim, na Avaliação, deve-se avaliar o aprendizado e a experiência do jogador. A execução dessas duas últimas fases serão detalhadas na Subseção 3.5.

3.4. Os objetivos de aprendizagem

O jogo está sendo desenvolvido visando contemplar o conteúdo programático das disciplinas introdutórias de programação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB) – *Campus* Campina Grande, 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. Dentre os tópicos citados, até a escrita do artigo, a primeira versão *beta* do jogo contempla os tópicos (1), (2) e (3).

O tópico (1) Raciocínio Lógico foi abordado por meio de desafios no estilo *puzzle* dentro do jogo, todos inseridos adequadamente no contexto e na narrativa do jogo. Como exemplo, tem-se o *puzzle* “empurra caixa” que é necessário ser resolvido para que o jogador possa obter um item importante do jogo (ver Figura 3a), e o *puzzle* “liga fios” que, quando resolvido, dá acesso à uma rede de comunicação (ver Figura 3b).

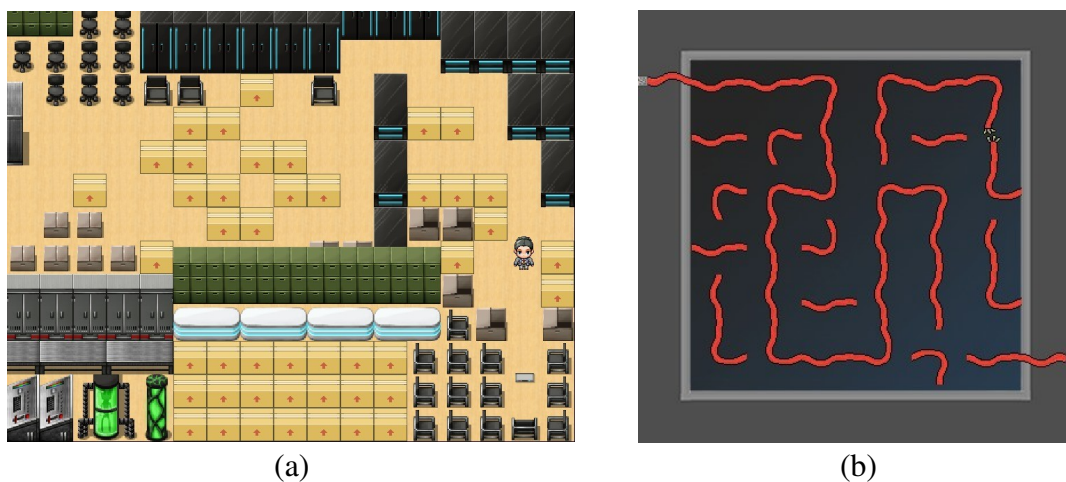


Figura 3. (a) *Puzzle* “empurra caixa” (b) *Puzzle* “liga fios”.

Os tópicos (2) Tipos de dados Python e (3) Operadores e expressões foram abordados por meio de pequenas tarefas de programação, as quais também foram inseridas cuidadosamente ao contexto e à narrativa de jogo. Essas tarefas estão basicamente relacionadas a objetos de Codeland que tiveram sua programação alterada pela IA. Assim, em cada tarefa de programação, o jogador é apresentado ao problema e ao código adulterado – com apenas uma das linhas de código modificada. Então, o

jogador deve solucionar o problema inserindo em uma caixa de texto a linha de código correta, e que vai corrigir o funcionamento do objeto. A linha de código inserida pelo jogador é processada pelo *plugin* desenvolvido, a fim de avaliar se o código está correto ou incorreto. No caso do código estar correto, a tarefa é completada e ele pode progredir no jogo; caso contrário, o jogador ficará impedido de progredir, mas poderá tentar tantas vezes quantas forem necessárias.

Dois exemplos das tarefas de programação são a “porta com senha” e o “pet robô”. No primeiro, o jogador se depara com uma porta que não abre, pois a IA alterou o código da porta que processa a senha de acesso. Para solucionar o problema, o jogador tem de lidar com *slicing* de string e concatenação. No segundo, o jogador tem que corrigir o código de um pet robô cujo senso de perigo não está bem ajustado. Nesse caso, o jogador tem de lidar com expressões lógicas. Ambas as tarefas podem ser vistas, respectivamente, nas Figuras 4a e 4b.

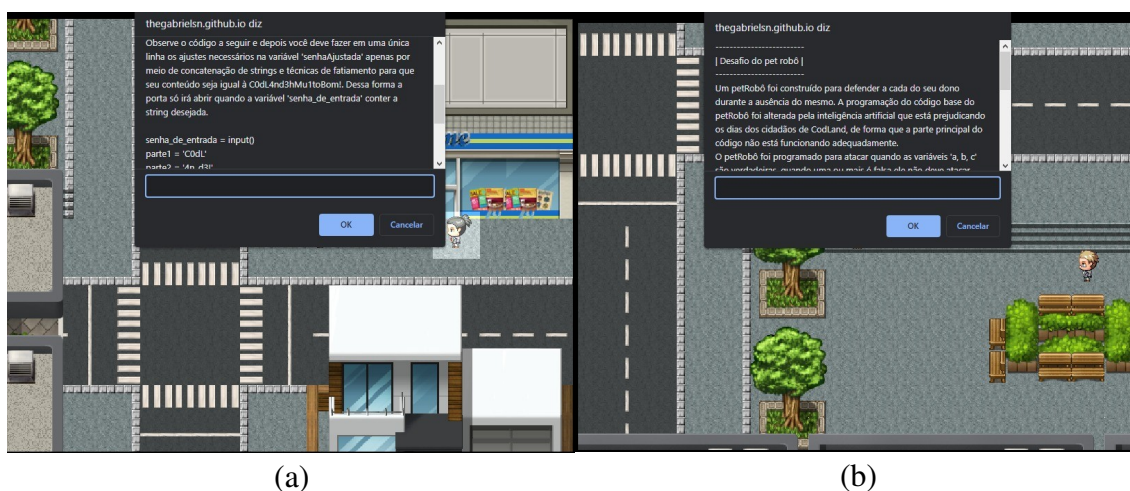


Figura 4. (a) Tarefa de programação “porta com senha” (b) Tarefa de programação “pet robô”

3.5. Avaliação da versão *beta*

Após a finalização da primeira versão *beta* de Codeland, iniciou-se a fase de Execução e Avaliação. Essas fases foram realizadas com a participação de 12 alunos voluntários que estavam cursando a disciplina de Introdução à Programação. Todos eram alunos do IFPB – *Campus* Campina Grande, matriculados nos cursos de Telemática ou Engenharia da Computação. Toda a pesquisa, o que incluiu a Execução e a Avaliação, foi realizada de forma remota em razão da pandemia da COVID-19. Os alunos voluntários receberam instruções precisas de como baixar, instalar e executar o jogo.

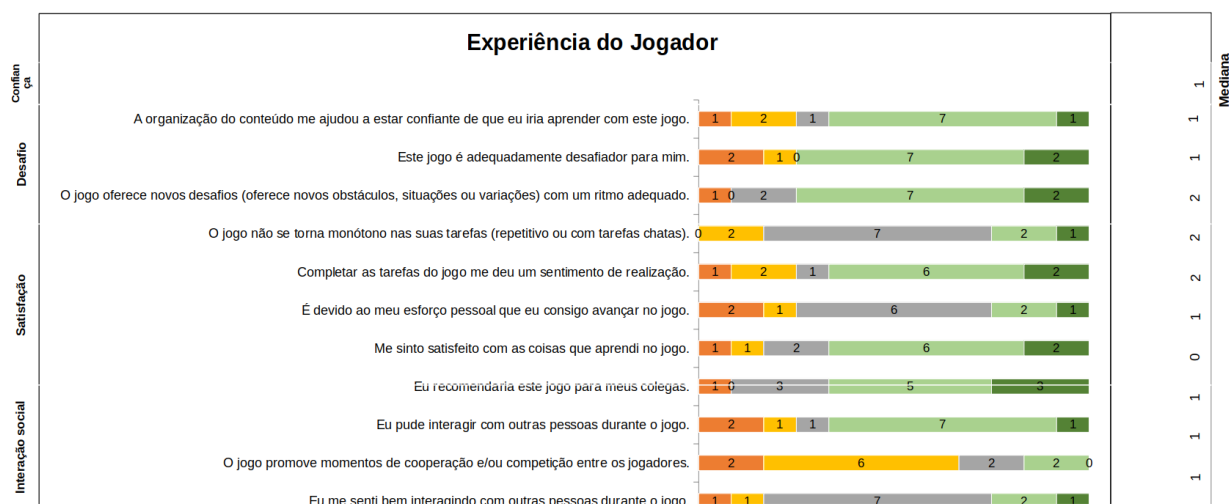
Após a experiência de jogar Codeland, os alunos foram submetidos ao questionário MEEGA+, um instrumento de medição para desenvolvedores a fim de avaliar a qualidade dos jogos educacionais [PETRI *et al.* 2019]. O modelo divide-se

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 têm 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.

Os aspectos concernentes à usabilidade de Codeland tiveram, em geral, muitas avaliações positivas. Os estudantes consideraram o jogo com uma estética atraente, fácil de jogar e com uma boa acessibilidade, sendo esta última a que obteve a melhor avaliação. Acredita-se que esses resultados tenham sido alcançados em função da utilização do RPG Maker MV, pois são aspectos inerentes à ferramenta.

As avaliações sobre a experiência do jogador podem ser vistas na Figura 5 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 impossibilidade de uma avaliação mais precisa frente à cobrança dos autores no retorno da avaliação em decorrência do avanço do tempo -, mas que não podem ser consideradas como respostas negativas. Um aspecto que se destacou foi o “*É claro para mim como o jogo se relaciona com a disciplina*”, com 83% de avaliações positivas. Acredita-se que esse resultado tenha sido alcançado em razão da utilização do processo ENgAGED, pois o processo guia o desenvolvimento do jogo de modo a se encaixar perfeitamente na metodologia da disciplina. Ademais, 83% consideraram o jogo como ferramenta adequada para a disciplina, muito embora 83% não tenham preferência para aprender com Codeland.



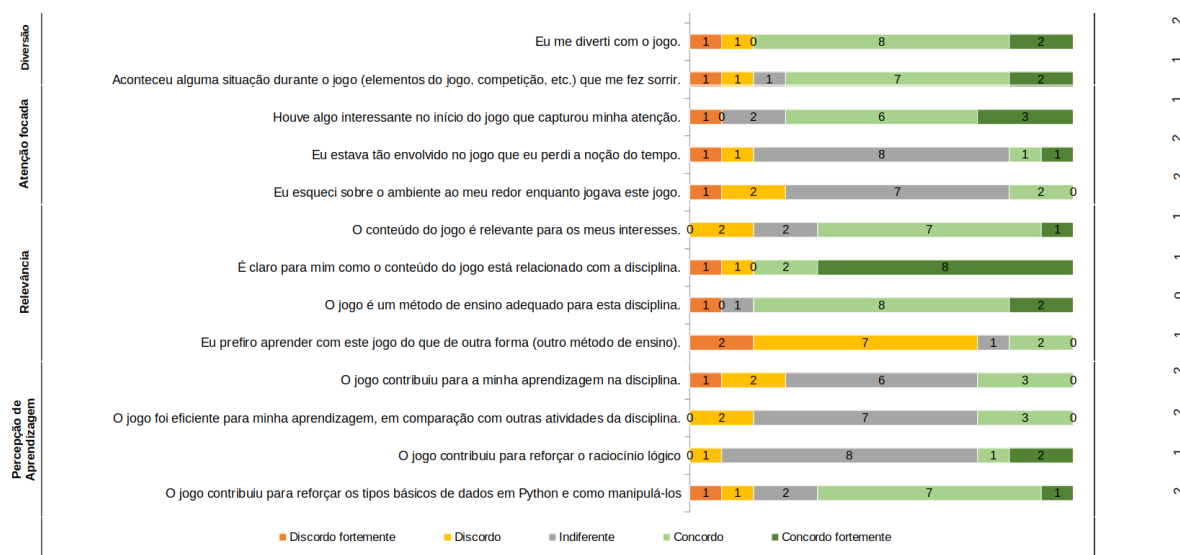


Figura 5. Resultado da avaliação da experiência do jogador.

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.

4. Considerações Finais

Este artigo apresentou a proposta do jogo educacional Codeland, uma ferramenta voltada para auxiliar alunos na aprendizagem de programação com Python em cursos superiores ligados à Computação.

Na avaliação realizada com os estudantes, através do questionário MEEGA+ após jogarem a versão *beta* do jogo, pôde-se observar tanto comentários positivos, como o caso do Estudante 1, que, ao se expressar, afirmou que “o jogo tem uma proposta muito massa, [...] uma maneira descontraída de aprender é muito interessante. [...] E isso ajuda o professor a fazer uma aula diferenciada. [...] Parabéns, a proposta é muito válida.”; quanto críticas construtivas, como o caso do Estudante 2 que sugeriu “melhorar a forma como a história é apresentada, pois está ocorrendo de forma rápida e direta.”

Em trabalhos futuros, pretende-se levar em consideração as sugestões dadas pelos alunos, as quais envolvem a correção de alguns *bugs* e a inclusão de mais tarefas de programação. Objetiva-se, ainda, dar continuidade ao jogo de modo a atender aos demais tópicos da disciplina de programação. Ademais, os autores intencionam definir os mecanismos para averiguar o quesito aprendizagem dos alunos que utilizarem o jogo.

Referências

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.

- Caspersen, M. E. e Kölling M. (2009) "STREAM: a first programming process". In: ACM Transactions on Computer Education, n 9, pp. 29.
- Esteves, M. *et al.* (2008) "Contextualization of Programming Learning: a virtual Environment study". In: Annual Frontiers In Education Conference, n. 38, Nova York. Proceedings, p. 17-22.
- Hartness, K. (2004) "RoboCode: using games to teach artificial intelligence". Journal of Computing in Small Colleges, n. 19, p. 287-291.
- Kadel, R. *et al.* (2019) "A Review on Educational Games Design, Development and Effectiveness Measurement". In: International Conference On Engineering, Technology And Education, n. 8. Yogyakarta. Proceedings, p. 1-7.
- 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.
- 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.
- Pimentel *et al.* (2019) "RPG Colaborativo para experimentação de vida acadêmica". In: Simpósio Brasileiro de Jogos e Entretenimento Digital, Anais SBGames, pp. 1216-1219.
- Rocha, V. R. (2017) "Critérios para a construção de jogos sérios". In: Congresso Brasileiro de Informática na Educação. Anais SBIE, p. 947-956.
- 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.
- Silva, R. R. *et al.* (2018) "Panorama da Utilização de Jogos Digitais no Ensino de Programação no Nível Superior na Última Década: Uma revisão Sistemática da Literatura". In: Congresso Brasileiro de Informática na Educação, n. 7, Fortaleza. Anais do SBIE. Editora SBC, p. 535-544.
- Steinmaurer, A. *et al.* (2019) "sCool - Game-Based Learning in Computer Science Class". International Journal of Engineering Pedagogy, n. 9, p. 35-50.
- Vahldick, A. *et al.* (2020) "A blocks-based serious game to support introductory computer programming in undergraduate education". Computers in Human Behavior Reports, n. 2, p. 1-12.
- Voulgari, I. *et al.* (2020) "Examining Student Teachers' Perceptions and Attitudes towards Game Based Learning". In: International Conference On The Foundations Of Digital Games, n. 20.. Proceedings. New York: ACM, p. 1-5.