

UNIVERSIDADE DE TAUBATÉ

Gleice Kelly dos Santos Moura

**USO DE JOGOS ELETRÔNICOS NO PROCESSO DE
ENSINO-APRENDIZAGEM NA DISCIPLINA DE LÓGICA DE
PROGRAMAÇÃO**

**Taubaté - SP
2020**

Gleice Kelly dos Santos Moura

**USO DE JOGOS ELETRÔNICOS NO
PROCESSO DE ENSINO-APRENDIZAGEM NA
DISCIPLINA DE LÓGICA DE PROGRAMAÇÃO**

Trabalho de Graduação Interdisciplinar,
apresentado como requisito parcial para a
conclusão do curso de Engenharia de
Computação do departamento de
Informática da Universidade de Taubaté.

Orientador: Prof. Me. Reuel Adimar Lopes.

**Taubaté - SP
2020**

Grupo Especial de Tratamento da Informação - GETI
Sistema Integrado de Bibliotecas – SIBi
Universidade de Taubaté - Unitau

M929u Moura, Gleice Kelly dos Santos

Uso de jogos eletrônicos no processo de ensino-aprendizagem na disciplina de Lógica de Programação / Gleice Kelly dos Santos Moura. -- 2020. 56 f. : il.

Monografia (graduação) – Universidade de Taubaté, Departamento de Informática, 2020.

Orientação: Prof. Me. Reuel Adimar Lopes, Departamento de Informática.

1. Jogos educativos. 2. Educação. 3. Programação (Computadores).
I. Universidade de Taubaté. Departamento de Informática. Graduação em Engenharia de Computação. III. Título.

CDD – 794.8

GLEICE KELLY DOS SANTOS MOURA

**USO DE JOGOS ELETRÔNICOS NO PROCESSO DE ENSINO-APRENDIZAGEM
NA DISCIPLINA DE LÓGICA DE PROGRAMAÇÃO.**

Trabalho de Graduação Interdisciplinar,
apresentado como requisito parcial para a
conclusão do curso de Engenharia de
Computação do departamento de
Informática da Universidade de Taubaté.

Data: _____

Resultado: _____

BANCA EXAMINADORA

Prof. Me. Reuel Adimar Lopes

Universidade de Taubaté

Assinatura _____

Profa. Ma. Alindacir Maria Dalla Vecchia Grassi Universidade de Taubaté

Assinatura _____

Prof. Me. Antonio Ricardo Mendrot

Universidade de Taubaté

Assinatura _____

Dedico este trabalho a minha família e amigos que me apoiaram durante
todos os anos de faculdade.

AGRADECIMENTOS

Meus agradecimentos ao meu orientador Prof. Me. Reuel Adimar Lopes por todo o apoio e interesse pelo meu trabalho. A Unitau e seus funcionários, principalmente os da biblioteca por nos proporcionar um ambiente de estudo e fontes de informação confiáveis.

Também agradeço a todos os que tiveram compreensão e paciência comigo durante o período de elaboração deste trabalho e que sempre estiveram presentes e dispostos a me ajudar.

Agradeço muito as pessoas que me proporcionaram o conhecimento necessário para concluir este trabalho, conhecidas e desconhecidas, presencialmente ou online. Pessoas que direta ou indiretamente contribuíram para meu aperfeiçoamento acadêmico, profissional e pessoal.

"O mundo as vezes pode parecer um lugar hostil e sinistro, mas acreditem: Existe muito mais bondade no mundo do que maldade, só precisam procurar com vontade. E o que pode parecer desventuras em série, na verdade é o primeiro passo para uma longa jornada."

Lemony Snicket.

RESUMO

A disciplina de lógica de programação requer habilidades de abstração e raciocínio lógico, as quais, para um aluno iniciante, pode ser difícil de aprender. A tecnologia vem sendo usada como facilitadora no processo de ensino por muitos anos e de muitas formas, por exemplo, o uso dos jogos eletrônicos. Neste trabalho, será abordado o uso dos jogos eletrônicos no ensino da disciplina de lógica de programação, características dos jogos e como podem oferecer benefícios ao processo de ensino-aprendizagem, incluindo aspectos relacionados à motivação, interação, ludicidade e ao protagonismo. Será apresentada revisão da literatura para compreender como um jogo pode contribuir para a construção de uma base sólida de conhecimento nesta disciplina. Também será apresentado um protótipo de um jogo digital que propõe a prática de lógica de programação, mesclando um desafio de lógica em algoritmo escrito em pseudocódigo com as vantagens e dinamismo de um jogo de aventura 2D linear.

Palavras-chave: Programação, Educação, Jogos educativos.

ABSTRACT

The subject of programming logic learning requires abstraction and logic reasoning abilities, and for a beginner student, those could be hard to learn. Technology has been used as a facilitator in the teaching-learning process for many years and in many ways, as for example by using computer games. In this paper, it approaches the use of electronic games in the subject of programming logic learning, games's characteristics and how it can be beneficial to learning, including among it's many aspects, motivation, interaction, playfulness and protagonism. It'll be presented a review of the literature on how a game could contribute to the construction of a solid knowledge foundation. It's also presented a prototype of a 2D digital game that provide practice of programming logic, mixing logic reasoning challenges in algorithms written in pseudocodes coupled with the advantages and dynamism of a 2D linear adventure game.

Keywords: Programming, Education, Educational games.

LISTA DE FIGURAS

Figura 1. Tabuleiro de Senet (MEGACURIOSO, 2020).....	15
Figura 2. Jogo eletrônico 'Space War' (Outerspace, 2020).....	15
Figura 3. Willy Higinbotham (OUTERSPACE, 2020).....	16
Figura 4. Captura de uma das fases do jogo "Code Combat".	28
Figura 5. Captura de uma das fases do jogo "Human Resource Machine".	29
Figura 6. Captura de uma das fases do jogo "7 Billion Humans".	29
Figura 7. Captura de uma das fases do jogo "Rabbids Coding".	30
Figura 8. Resumo do conteúdo do "2D Game Kit"	32
Figura 9. UI inicial do jogo, parte do "2D Game Kit".	33
Figura 10. Exemplo de cenário do "2D Game Kit".	33
Figura 11. Introdução da história por meio de infopost.	36
Figura 12. Protagonista encontra uma chave de segurança.	37
Figura 13. Diário pertencente à protagonista, aba Diagrama.	38
Figura 14. Diário pertencente à protagonista, aba Teoria.	38
Figura 15. Terminal pertencente à protagonista.	39
Figura 16. Diagrama de Chapin. Retirado do livro "Curso essencial em lógica de Programação", 2008.....	40
Figura 17. Cena 1.....	42
Figura 18. Cena 2.....	42
Figura 19. Diagrama com o algoritmo corrigido, desafio 1.	43
Figura 20. Código com falha, desafio 1.....	44
Figura 21. Visão hierárquica da UI Diário.....	44
Figura 22. Excerto do script DiaryNav.cs	45
Figura 23. Visão hierárquica da UI Terminal	46
Figura 24. Inspector do objeto Chave_VAR	46
Figura 25. Inspector do objeto Var_1	47
Figura 26. Inspector do Content_terminal com KeyCodeEnable.....	47
Figura 27. Excerto do script KeyCodeEnable.cs	47
Figura 28. Inspector do objeto Executarbutton.....	48
Figura 29. Inspector do Content_terminal com TerminalExecute.....	48
Figura 30. Excerto do script TerminalExecute.cs	49
Figura 31. Inspector do game object Key	49

LISTA DE TABELAS

TABELA 1. Elementos integrantes de jogos eletrônicos	20
--	----

SUMÁRIO

1. INTRODUÇÃO	11
1.1. OBJETIVO GERAL	12
1.2. OBJETIVO ESPECÍFICO	12
1.3. JUSTIFICATIVA	12
2. JOGOS ELETRÔNICOS	14
2.1. CONCEITO DE JOGO	14
2.2. HISTÓRICO	15
2.3. JOGOS ELETRÔNICOS EDUCATIVOS	17
2.4. CARACTERÍSTICAS E ELEMENTOS GERAIS	19
2.5. ELEMENTOS MOTIVACIONAIS	21
3. JOGOS ELETRÔNICOS NO ENSINO DE LÓGICA DE PROGRAMAÇÃO	24
3.1. APRENDIZAGEM SIGNIFICATIVA NO APRENDIZADO DE RESOLUÇÃO DE PROBLEMAS	26
3.2. EXEMPLOS DE JOGOS PARA ESTUDO DE LÓGICA DE PROGRAMAÇÃO	27
3.3. EFETIVIDADE NO USO DE JOGOS ELETRÔNICOS EDUCATIVOS	30
4. DESENVOLVIMENTO	31
4.1. FERRAMENTAS E COMPONENTES	31
4.1.1. ENGINE	31
4.1.2. ASSETS	32
4.1.3. FERRAMENTAS GRÁFICAS	33
4.2. METODOLOGIA	34
4.3. ESTRUTURA DO PROTÓTIPO	35
4.3.1. JOGABILIDADE E REGRAS	36
4.3.2. CULTURA: REPRESENTAÇÃO DE ALGORITMO	39
4.4. PROTÓTIPO DO JOGO “THE EXPLORER”	41
4.4.1. NÍVEL	41
4.4.2. SCRIPTS	44
5. CONCLUSÃO	50
REFERÊNCIAS	51
APÊNDICE A – GAME DESIGN DOCUMENT	55

1. INTRODUÇÃO

Uma das disciplinas mais importantes dos cursos de computação é a lógica de programação. É através do seu domínio que se faz possível aprender conceitos mais complexos, a fim de criar aplicações para solução de problemas do mundo real. Sendo assim, é grande a importância de se obter uma base sólida de conhecimento nessa disciplina.

Sendo uma disciplina introdutória, entende-se que os alunos ainda não desenvolveram a habilidade de pensar de forma lógica e sistemática. É do interesse do professor que essas dificuldades não sejam causa de desistência ou reprovação. Conceitos abstratos e complexos requerem prática e estudo para que sejam bem compreendidos, e muitas vezes os alunos se sentem desmotivados por não serem capazes de dominar esses conceitos rapidamente (IBRAHIM et al, 2011).

Para manter o aluno motivado e engajado, são utilizados métodos e ferramentas diferenciados, e um deles é o jogo eletrônico. Os jogos eletrônicos são fontes de diversão e entretenimento, porém seu valor como ferramenta educativa vem sendo reafirmado por diversos autores nos últimos anos.

Segundo Haguenauer (2007), os jogos auxiliam no processo de aprendizagem “facilitando o desenvolvimento de novas habilidades como observar e identificar, comparar e classificar, conceituar, relacionar e inferir além de desenvolver a criatividade, perseverança e sociabilidade”.

Gomez (2010, p. 09 apud SILVA, 2016) defende a difusão da tecnologia na cultura e na formação das pessoas no mundo contemporâneo ao afirmar que:

Estamos caminhando para uma fase de convergência e integração das mídias. Tudo começa a integrar-se com tudo e falar com tudo e com todos. Há um diálogo crescente, muito novo e rico entre o mundo físico e o chamado mundo digital, com suas múltiplas atividades de pesquisa, lazer, de relacionamento e outros serviços e possibilidades de integração entre ambos, que impactam profundamente a educação escolar e as formas de ensinar e aprender a que estamos habituados. As mudanças que estão acontecendo na sociedade, mediadas pelas tecnologias em rede, são de tal magnitude que implicam – a médio prazo – em reinventar a educação como um todo, em todos os níveis e de todas as formas. (SILVA, 2016, p.349)

O trabalho é composto de 5 capítulos, sendo este o primeiro. No capítulo 2 é apresentado: conceito de jogo, breve histórico, suas características gerais e motivacionais, assim como benefícios dos jogos educativos. No capítulo 3 será

abordada a forma como jogos podem auxiliar no ensino de lógica de programação especificamente, e apresentados alguns exemplos de jogos digitais para o ensino dessa disciplina. No último capítulo será apresentado o protótipo desenvolvido a fim de sugerir a prática de conceitos de lógica de programação e familiarização com formas de representação de algoritmos.

1.1. Objetivo Geral

Este trabalho tem por objetivo compreender como os jogos eletrônicos podem contribuir com o processo de ensino-aprendizagem na disciplina de lógica de programação.

1.2. Objetivo Específico

- Apresentar uma revisão de literatura, abrangendo os seguintes tópicos: principais conceitos e características dos jogos com foco educacional; jogos educacionais para ensino de programação, e exemplos de jogos educativos para o ensino de programação.
- Desenvolver protótipo de jogo para a prática dos conceitos básicos de lógica de programação.

1.3. Justificativa

A disciplina de lógica de programação apresenta conceitos abstratos e complexos que requerem prática e estudo para que sejam bem compreendidos, e muitas vezes os alunos se sentem desmotivados por não serem capazes de dominar esses conceitos rapidamente. Por esse motivo, é importante buscar novas abordagens e ferramentas que facilitem o entendimento e a prática do conteúdo dessa

disciplina. Assim, espera-se que este trabalho contribua para a compreensão dos benefícios provenientes com o uso dos jogos eletrônicos no processo de ensino-aprendizagem da disciplina.

2. JOGOS ELETRÔNICOS

2.1. Conceito de Jogo

Os jogos sempre fizeram parte do cotidiano das pessoas, mesmo antes da invenção dos jogos eletrônicos. Jogar vem de maneira natural, para entreter e/ou para aprender, tudo pode ser transformado em um jogo. Sendo assim, conceituar, definir o que vem a ser um jogo, deve levar em conta todas as transformações que eles sofreram através da história, todas as variações nas formas de jogar e todas as finalidades a que se destina cada tipo de jogo. Essa complexidade intrínseca à existência dos jogos resulta em definições variadas, das mais técnicas às mais filosóficas. “O jogo é de fato mais antigo que a cultura” afirmou Huizinga em *Homo Ludens* (2000).

Brian Sutton-Smith (apud Salen & Zimmerman, 2004) afirma que cada pessoa define os jogos da sua própria maneira: os antropólogos em termos históricos, os executivos e os educadores em termos práticos, os sociólogos em termos psicológicos e sociais, ou seja, os jogos são para cada pessoa manifestações de suas próprias ideias.

Salen & Zimmerman (2004) ainda complementam afirmando que, como produto da cultura humana, seria ineficaz tentar enxergar um fenômeno tão complexo através de uma única perspectiva. Assim, embora complicado de definir, alguns autores propuseram alguns conceitos.

David Parlett define o jogo formal (excluindo-se o simples ato de brincar) em termos de meios e finalidades. Para ele, a finalidade do jogo é conquistar um objetivo, já o meio é o conjunto de regras e materiais usados para conquistar esse objetivo (SALEN & ZIMMERMAN, 2004).

Neste trabalho, fora adotada a definição de jogo formal, definido por David Parlett, por apresentar parâmetros mais claros e objetivos se comparado a outras definições. Logo, a conquista do objetivo e a adoção de regras são aspectos fundamentais na composição de qualquer jogo.

2.2. Histórico

Segundo Costa (2014), a existência dos jogos remonta a épocas antigas, como visto no jogo Senet, que é datada em 3.500 a.C. e que possuía tanto a funcionalidade de entreter como a de educar, a Figura 1 mostra o tabuleiro físico utilizado para jogar Senet.



Figura 1. Tabuleiro de Senet (MEGACURIOSO, 2020).

O primeiro jogo eletrônico interativo criado em computador foi o chamado “Space War”, criado em 1961 por um estudante do MIT, Steve Russell. Antes disso, Willy Higinbotham programou, em 1958, um osciloscópio onde era possível jogar uma partida de tênis interativa (KISHIMOTO, 2004). A Figura 2 mostra uma tela do jogo.



Figura 2. Jogo eletrônico ‘Space War’ (Outerspace, 2020)



Figura 3. Willy Higinbotham (OUTERSPACE, 2020)

Nos anos 70, as empresas de jogos, como Taito, Midway e Capcom, apostaram na tecnologia e desenvolveram versões comerciais de máquinas capazes de executar os jogos, assim como desenvolveram novos jogos, como fliperamas e alguns computadores (KISHIMOTO, 2004).

Já os anos 80, houve popularização dos consoles de videogame de 8-bits, como o Famicom e o NES, da Nintendo, e na concorrência o Mark III e o Master System, da SEGA. De 80 para 90, a Nintendo lança o Game Boy, um portátil em preto e branco, e o Super NES (HORTA & SILVA, 2011).

A competitividade entre as empresas de videogame nos anos 90 fez com que os consoles se aperfeiçoassem constantemente. Houve o Sega Genesis e Super Famicom, da Nintendo, de 16-bits, o PlayStation, da Sony, e Sega Saturn (32 bits) e o lançamento de videogames como 3DO, da Panasonic, e Nintendo64, da Nintendo (64-bit). Na passagem para 2000, lançaram-se consoles em 128-bits, a Sony com o PlayStation 2 e a Nintendo com o GameCube (KISHIMOTO, 2004).

No século XXI surgiram e continuam a surgir consoles cada vez mais avançados, como a série Playstation, Xbox e Nintendo Wii, assim como outras modalidades de jogos, como os jogos de computador on-line multiplayer massivos (HORTA & SILVA, 2011).

2.3. Jogos eletrônicos educativos

A presença das Tecnologias de Informação e Comunicação (TIC) na área da educação tem crescido consistentemente nos últimos anos. Isso porque o apoio do governo tem base nos objetivos de criação de uma cultura digital, entre eles promover a autonomia dos alunos e dinamismo em sala de aula (BIELSCHOWSKY, 2009).

O estudo “ICT Impact Report” (BALANSKAT et al apud BIELSCHOWSKY, 2009) afirma que o uso de TIC é benéfico ao professor, pois “permite uma maior diferenciação entre os estudantes, com programas mais adaptados para as suas necessidades individuais” e para os estudantes, pois “os estudantes assumem uma maior responsabilidade no seu processo de aprendizagem com o uso de TIC, trabalhando de forma mais independentemente e eficiente”.

Segundo a pesquisa TIC Educação de 2017, os jogos eletrônicos podem contribuir “para identificar o estágio de desenvolvimento cognitivo no qual o aluno se encontra, sinalizando ao professor como intervir para potencializar o processo de aprendizagem do estudante”.

Jogos eletrônicos educativos utilizam os elementos de jogos em plataforma digital para transmitir informações e facilitar o aprendizado. Dentro dessa definição, existem desde jogos de entretenimento com benefícios para aprendizagem até jogos com propósito educacional, os jogos sérios. Depende de cada designer e equipe de desenvolvimento definir as características e objetivos de seu jogo (SALEN & ZIMMERMAN, 2004).

Jogos sérios, em inglês, *serious games*, são popularmente utilizados em aulas e treinamentos. Suas aplicações têm sido eficientes nas áreas de saúde, defesa, negócios, turismo, entre outras (ALDRICH, MATTAR e NOVAK apud ROCHA & BITTENCOURT & ISOTANI, 2015).

Entre 2016 e 2017, 1718 jogos foram desenvolvidos no Brasil, destes 874 foram jogos sérios. Em 2016 foram 411 e, em 2017, 463 jogos sérios foram desenvolvidos no Brasil, com um crescimento de 13% (SAKUDA, 2018).

Segundo Rocha, Bittencourt e Isotani (2015), o desafio em desenvolver jogos educativos é balancear os requisitos do jogo, como entretenimento, com aspectos mais sérios e técnicos de simulação e conteúdo educativo, a fim de alcançar objetivos,

como aprendizagem ativa, prática das competências, motivações, objetivos e pré-requisitos claros, através de vários cenários e problemas propostos.

Para Hernández-Ramos et al. (2016), o diferencial do jogo como ferramenta educativa é que o processo de aquisição de conhecimento e aperfeiçoamento de habilidades são reflexos do engajamento do aluno com os problemas e desafios propostos a fim de que, a cada etapa concluída, sua motivação seja renovada através do sentimento de conquista. Essa afirmação é corroborada por Otsuka e Beder (2018), pois afirmam que o estudo e prática dos conhecimentos adquiridos são reconhecidos pelos alunos como uma estratégia para que possam ganhar o jogo ao invés de uma obrigação sem um objetivo claro, transferindo para o aluno certo controle e responsabilidade no processo de aprendizagem.

Autores citam diversas vantagens no uso do jogo no processo de ensino-aprendizagem, por exemplo a construção da autoconfiança e autonomia, quando o aluno passa a ser capaz de resolver problemas independentemente. O fator de ludicidade oferece um ambiente mais leve e descontraído, fazendo com que o aluno tenha menos medo de errar e, assim, possa explorar melhor o conteúdo (BARBOSA, 2011; BIELSCHOWSKY, 2009; WERNER & GONÇALVES, 2016).

Costa (2014) relacionou alguns diferenciais oferecidos pelos jogos eletrônicos:

- Interatividade imediata e restrita, em que as ações dos jogadores resultam em resposta quase sempre instantânea e específica;
- Manipulação da informação: é possível dosar a quantidade e a ênfase de determinado conteúdo de acordo com as necessidades de aprendizagem de forma a integrá-lo ao jogo para que esse processo se repita automaticamente.
- Sistemas complexos automatizados: com o uso da tecnologia é possível desenvolver jogos complexos, mas de fácil acesso.
- Comunicação via Internet: embora nem todos os jogos sejam multiplayer, há comunidades *gamers* que possibilitam aos alunos interação e debate entre si.

Costa (2014) ainda destaca o trabalho de Seymour Papert ao relacionar o construtivismo com a prática de jogos eletrônicos educativos, afirmando que “apesar do papel fundamental da tecnologia na educação, o foco não está na máquina, e sim na mente do usuário”. O autor propõe que o jogo auxilia no trabalho do professor que,

baseado na teoria construtivista, não impõe o seu saber ao aluno, mas o acompanha, incentiva, sugere e aprende junto.

2.4. Características e elementos gerais

Jogos eletrônicos educativos são sistemas complexos, não somente no quesito tecnologia, mas também em relação à abordagem pedagógica. São muitas as variáveis, porém existem certas características comuns à maioria dos jogos.

Salen e Zimmerman, em seu livro *Rules of Play*, discutem os aspectos necessários para se desenvolver um jogo. Eles observaram que, para construir um jogo que vai além do superficial, uma pessoa no passado teria que ter conhecimentos de diversas disciplinas, como sociologia, antropologia, psicologia e matemática. Sendo assim, o profissional comum não tem domínio de tantas disciplinas e, mesmo assim, ainda há a questão de como conectar a teoria com os elementos concretos do jogo. O ato de criar um jogo vai além do conhecimento e da habilidade, pois é um mundo de infinitas possibilidades, sendo a criatividade o papel principal. Os autores então propõem uma questão: o que o desenvolvedor quer que o jogador possa fazer e experienciar enquanto está controlando seu personagem dentro desse mundo simulado? Uma vez encontrado seu propósito, o design do jogo definirá características que sejam úteis e complementares a ele.

Quanto aos aspectos computacionais e de outras ciências, Clua e Bittencourt (2005) afirmam que o design de jogo é uma área multidisciplinar e envolve profissionais das áreas tais como educação, psicologia, artes plásticas, letras, design gráfico e música, além de ciência da computação principalmente nas disciplinas de análise de algoritmos (otimização), computação gráfica, redes de computadores e inteligência artificial.

Em *Rules of Play* são realizadas análises de aspectos comuns dos jogos de sucesso, mostrando que aspectos como a interatividade, representação e a socialização são característicos de jogos eletrônicos, pois são a base para a imersão do jogador no mundo criado, necessário para que ele experiencie o máximo do que o jogo tem a oferecer. Quanto mais engajado o jogador estiver, maiores são as chances de que seu propósito seja cumprido.

Rocha et al (2015), apontam que os elementos de um jogo educativo devem atender aos princípios de aprendizagem ativa (prática de competências, motivação, objetivos claros, simulações entre outros), mantendo um equilíbrio com as outras características do jogo. Para isso, propõem que pontos fortes da educação, como teoria de aprendizagem e métodos de ensino, sejam aplicados ao design do jogo, assim como habilidades de diferentes áreas, como o gerenciamento de projetos, por exemplo, e que, ao desenhar o jogo, suas características e estratégias sejam alinhadas com os objetivos da aprendizagem. É indispensável que se faça claro na etapa de planejamento qual é o público-alvo, o conteúdo a ser abordado e como ocorrerá a avaliação do desempenho.

Ademais, Costa (2014) lista, em seu trabalho sobre gamificação, de 2014, alguns aspectos comumente empregados nos jogos, conforme descrito na tabela 1.

Tabela 1. Elementos integrantes de jogos eletrônicos. (Adaptado da dissertação de mestrado 'Ambiente de aprendizagem de estruturas de dados usando gamificação'. COSTA, 2014).

Elemento de jogo	Descrição
Alvos	Um alvo pode ser considerado como um objetivo fixo, que pode ser quantificável e é facilmente percebido pelo usuário. Estes podem ser usados para criar objetivos de curta duração ou de longa duração, direcionando as ações do usuário no ambiente.
Competição	Este elemento é uma manifestação natural de rivalidade entre usuário ou grupos de usuários quando há conflito pelo mesmo objetivo.
Chance	Chamado também de aleatoriedade ou sorte, este é um dos mais antigos elementos presentes no cotidiano da humanidade.
Pressão por Tempo	O tempo é algo fundamental na realidade dos usuários.
Escassez	A escassez é definida como a falta de recurso. Quando ocorre esta falta, a demanda daquele recurso tende a aumentar. A sua utilização faz com que o usuário seja mais cuidadoso com o que possui.
Puzzles	Este elemento representa problemas que possuem uma solução, e cabe ao usuário descobri-la.
Nível	Organizado de forma hierárquica e vinculada ao domínio abordado, os níveis produzem uma sequência lógica para a obtenção de habilidades de forma gradual e organizada.
Pressão Social	Este elemento é gerado a partir da interação entre os indivíduos. Outra característica marcando é sentimento que se a maioria faz, está certo. Um ambiente com pouca

	pressão social acaba fazendo com que o usuário sinta pouca obrigação em usar aquele ambiente.
Trabalho em Equipe	Quando mais de um jogador trabalham juntos para atingir um objetivo em comum, tem-se o que é chamado de trabalho em equipe.
Moeda Corrente	Assim que se encontra a necessidade de realizar transações comerciais dentro do jogo, a criação de uma moeda corrente se faz necessária.
Renovação	Este elemento representa o processo de recomeçar. O processo de tentativa e erro é necessário existir quando há um conceito muito difícil sendo abordado.
Julgamento	Os julgamentos são aqueles que o jogador precisa tomar para continuar uma determinada atividade.
Dados	Um dado pode ser caracterizado como tal quando uma informação é apresentada ao jogador.
Progresso	Este elemento é um dado especializado em mostrar ao jogador o quão longe ele percorreu e o quanto falta para encerrar o jogo, ou a atividade. A falta de retorno ao jogador de quanto ele progrediu acaba deixando-o ansioso.
Sensação	Pelo fato de que os jogadores são indivíduos, estes tendem a receber forte influência de suas emoções.
Reconhecimento	O reconhecimento é toda conquista alcançada e percebida pelo jogador e por outros. A falta de reconhecimentos dentro de um ambiente acaba diminuindo a motivação do usuário e tende a aumentar o sentimento de isolamento do usuário.
Status	O Status pode ser considerado como a manifestação do poder e respeito em um grupo social.

2.5. Elementos motivacionais

Peres (2016) realizou uma análise da literatura referente a elementos motivacionais de jogos eletrônicos, principalmente os trabalhos realizados por Sweetser et al. e Weibel e Wissmath, a fim de elencar os principais elementos de sucesso de um jogo. Os melhores 5 elementos dessa análise foram, em ordem de maior importância: Publicação, Fluxo, Imersão, Recompensa e Cooperação.

O autor ainda fez uma segunda análise usando outro método de investigação próprio. Os 5 elementos principais identificados pelo autor foram, em ordem de importância: Cooperação, Status, Fluxo, Competição e Imersão.

- Publicação se refere à capacidade de compartilhar o jogo em si, assim como suas conquistas.
- Competição implica sempre em um perdedor e em um vencedor.
- Fluxo permite que o jogador mantenha o foco e a concentração, mantém o usuário jogando por mais tempo. Muito fácil, entediado. Muito difícil, frustrado.
- Imersão é definida como a capacidade do jogo de fazer o jogador se sentir parte do mundo fictício.
- Cooperação pode ser definida como a possibilidade de colaborar com outras pessoas no jogo, muitas vezes necessária para passar de nível.
- Status representa a sensação de poder obtida pelo jogador, isto é, a sensação de que o seu progresso no jogo o torna superior a outros jogadores de alguma forma.
- Recompensa representa algo que o jogador ganha pelos obstáculos e desafios vencidos.

Outro aspecto importante na construção do jogo é apontado por Sousa (2017) em seu estudo sobre jogos significativos: o Flow, em português Fluxo.

Há teorias científicas de que jogos que promovem um estado de fluxo aumentam o nível de motivação dos jogadores. Fluxo é um estado de profundo prazer e concentração em situações em que as habilidades de alguém estão equiparadas com a dificuldade do desafio (GARRIS et al apud HERNÁNDEZ-RAMOS et al, 2016).

Sousa (2017) indica alguns aspectos que auxiliam na manutenção do estado de fluxo, são eles: objetivos claros, alto nível de concentração, alto nível de imersão, sensação de tempo distorcida, feedback imediato, equilíbrio entre dificuldades nos desafios propostos, sensação de controle sobre a atividade, e atividade prazerosa.

Hernández-Ramos et al. (2016) revisaram pesquisas nesta área e concluíram que a interação com o jogo aperfeiçoa a curva de aprendizado e inclui imersão no ambiente, absorção prazerosa através do estado de fluxo e da identificação com o personagem no jogo.

Em referência ao impacto causado pelo fluxo no desempenho do aluno, Hernández-Ramos et al. (2016) analisaram que os resultados ainda são inconclusivos. Em estudos recentes foram observados casos em que o estado de fluxo teve impacto

positivo em crianças, enquanto outros estudos mostraram que essa abordagem não tem o mesmo impacto para todo tipo de ensino. Os autores usam como exemplo os estudos de Admiraal, Huizenga, Akkerman, e Dam (2011), no qual concluíram que o desempenho em grupo num jogo de história medieval foi bom, porém ruim em uma avaliação conceitual.

3. JOGOS ELETRÔNICOS NO ENSINO DE LÓGICA DE PROGRAMAÇÃO

Segundo Almeida (2008), a lógica de programação tem como objetivo usar métodos a para definição de uma sequência lógica de instruções computacionais para o desenvolvimento de um programa com a finalidade de resolver algum problema. Para isso, são desenvolvidos algoritmos que devem ser descritos de acordo com regras sintáticas e semânticas específicas para cada linguagem de programação. Almeida (2008) ainda argumenta que a aplicação da linguagem de programação é simples de ser aprendida e requer somente conhecimento de suas regras, enquanto a parte essencial da lógica de programação é o algoritmo a ser traduzido para a linguagem escolhida.

Algoritmo, segundo o dicionário Michaelis, é o “conjunto das regras de operação (conjunto de raciocínios) cuja aplicação permite resolver um problema enunciado por meio de um número finito de operações”. Said (2007) reafirma essa definição e a expande: “conforme precisemos especificar uma sequência de passos, é necessário utilizar ordem, ou seja, ‘pensar com ordem’, utilizar lógica.” Dessa forma, é necessário abordar o problema de forma sistemática.

Uma habilidade fundamental para a construção de algoritmos é a capacidade de resolução de problemas. A fim de definir corretamente a sequência lógica do algoritmo, é necessário abstrair da situação real dados que permitam analisar e manipular a situação-problema para que seja alcançada a solução desejada. Após a identificação da solução e variáveis envolvidas, é possível construir o algoritmo. Há muitos métodos para isso, sendo os mais populares: a. descritivo, através do uso de linguagem natural não estruturado, como numa narrativa; b. pseudocódigo, uso de linguagem natural estruturada simples; c. representativo, por meio de diagramas, fluxogramas etc. (ALMEIDA, 2008; BARBOSA, 2011; BARCELOS, 2012; RAPKIEWICZ et al, 2006; SAID, 2007).

Muitos estudos destacam a deficiência no ensino de lógica e resolução de problemas como um dos motivos principais de evasão e reprovação em cursos de computação. Outros motivos são: a falta de experiência anterior com a disciplina; falta de acompanhamento do professor; didática expositiva e pouco colaborativa; desmotivação por não conseguir perceber o próprio progresso ou percepção de que

a disciplina é difícil demais (BARBOSA, 2011; NUNES, 2016; RAPKIEWICZ et al, 2006).

Algumas características apontadas como crucial para o aperfeiçoamento das habilidades de bons programadores são: atenção, organização, concentração, perseverança, padronização, pensamento criativo e otimização. Neste quesito, o aluno pode beneficiar-se substancialmente ao utilizar jogos educativos que promovem a prática da lógica e da construção de algoritmos. (ALMEIDA, 2008; RAPKIEWICZ et al, 2006).

Quanto a outra habilidade importante para a disciplina, a abstração, Moratori (2003) afirma que “o jogo depende da imaginação e é a partir desta situação imaginária que se traça o caminho à abstração”.

Segundo Rapkiewicz et al (2006), a “teoria dos Jogos consiste na codificação do pensamento estratégico” e afirma:

Embora o “pensamento estratégico” receba críticas – o ato de ganhar é bem definido e facilmente discernido. Fayard (2000) enfatiza que a liberdade de ação e a perspectiva de incerteza do outro podem se tornar motivação para melhorar a situação do ator no conflito. O jogador visa alcançar uma meta, que é ganhar o jogo, e por conta de seus objetivos usa estratégias. [...] É fundamental, pois, associar ao recurso do jogo um conjunto de estratégias que permita ao professor, como mediador do processo de ensino e aprendizagem, conduzir o aluno a construir sua própria solução e representá-la na forma de algoritmos (RAPKIEWICZ, 2006).

Souza (2017), distingue entre duas modalidades de jogos que podem ser utilizados de forma educativa: a. jogos educativos, com objetivo principal de ensinar e distribuir conhecimento, como os jogos de simulação US Army e Flight simulator; e b. os de entretenimento, que foram desenvolvidos com o objetivo principal de entreter mas que possuem potencial educativo, como o Spore – para ensino de biologia, por exemplo.

Rapkiewicz et al. (2006) propõem que os jogos podem ser utilizados no ensino de algoritmos, mesmo que não explicitamente, ao propor que a lógica dos jogos pode ser em si analisada e desconstruída. Os desafios propostos pelo jogo a serem superados podem ser descritos em forma de algoritmo, como exercício proposto.

Também é possível, por meio de jogos de simulação, propor problemas puros de lógica para desenvolver as habilidades de resolução de problemas.

3.1. Aprendizagem significativa no aprendizado de resolução de problemas

Phye (1997), discursa sobre diversos métodos de ensino e suas características. Nele, é debatido o mérito de cada método de acordo com o objetivo de aprendizagem, ou seja, a finalidade para a qual se deseja aprender algo.

Referente à aprendizagem de resolução de problemas, Phye (1997) faz algumas distinções entre resolução rotineira – em que a aprendizagem mecânica é o suficiente para sua resolver o problema, sendo imperativo somente a habilidade de retenção de conteúdo – e resolução criativa, quando o problema que se apresenta é novo e ainda não se sabe como resolvê-lo, sendo necessária a habilidade de aplicação do conteúdo aprendido.

Sem dúvida, a aprendizagem de lógica de programação de modo eficaz exige resoluções criativas, pois o programador, no exercício de sua função, terá que solucionar problemas para os quais ainda não conhece a solução.

Phye (1997) argumenta ainda que métodos de aprendizagem significativa são mais adequados para ensino de conteúdo que requer aplicação – nos quais se procura a produção de resolução criativa – do que métodos tradicionais de aprendizagem mecânica.

A aprendizagem significativa se baseia na visão construtivista teorizada por David P. Ausubel, em 1968, de que a aprendizagem de um conhecimento novo é influenciada por um conhecimento anterior. Em outras palavras, nenhum conhecimento é compreendido de maneira isolada, pois a mente humana irá criar conexões entre o que aprendeu e o que já sabia, sendo mais eficaz compreender novos conceitos se estes estão ligados a conceitos que já são conhecidos e experienciado. Da mesma forma, quando conceitos são aprendidos incorretamente, esse conhecimento irá influenciar de forma negativa a compreensão de novos conceitos. Por esse motivo, adeptos da aprendizagem significativa defendem que é

crucial que, ao assimilar novos conceitos, o aluno seja motivado a revisar e colocar à prova conhecimento anteriores (PHYE, 1997; SOUZA, 2017)

Fica claro, portanto, como a aprendizagem significativa é importante no aprendizado de lógica de programação, em que cada conceito novo é uma peça na construção do algoritmo, então sua correta compreensão irá possibilitar o entendimento de conceitos mais complexos que, em conjunto, são utilizados durante o processo de resolução do problema.

Rapkiewicz et al (2006) argumenta que o uso dos jogos oferece uma vantagem nesse quesito, pois “através da simulação o aluno consegue visualizar os passos necessários para alcançar um resultado”, e que o jogo “permite tornar real o que seria apenas imaginário no caso de enunciados de problemas apresentados no quadro ou em listas de exercício em papel”. Desta forma, é possível integrar tanto conceitos de programação dentro dos jogos como criar conexões entre eles de forma dinâmica.

Souza (2017) levanta outros dois aspectos dos jogos que facilitam a aprendizagem significativa: motivação dos alunos em aprender por conta própria e material ou ferramenta de aprendizado significativo para o aluno. Assim sendo, a utilização de jogos pode causar engajamento maior dos alunos por ser algo com o que já estão familiarizados, sendo visto de forma positiva e divertida.

3.2. Exemplos de jogos para estudo de lógica de programação

Em “A Review of Serious Games for Programming” (2000), Miljanovic e Bradbury reuniram 49 jogos sérios para programação de computadores e os categorizaram por público-alvo e áreas de conhecimento. O maior público foi “estudantes universitários iniciantes sem experiência em programação” (21 jogos), e a área de conhecimento principal foi “conceitos fundamentais em programação” (48 jogos) – mas alguns jogos puderam ser classificados em mais de uma área de conhecimento. “Algoritmos e Design” e “Estruturas de dados fundamentais”, que exploram a resolução de problemas e estruturas como vetores e matrizes, respectivamente, foram incluídos, cada um deles, em 23 dos jogos pesquisados. A seguir a lista de jogos citados no trabalho de Miljanovic e Bradbury: Minecraft: Hero’s Journey, Code Combat, ToonTalk, PlayLogo 3D, Software KIDS, Cquest, World of

Variables, Unnamed RPG, May's Journey, Co.Co.I.A., RoboBuilder, Super Markup Man, Human Resource Machine, Unnamed Maze, Unnamed Puzzle, Wu's Castle, BOTS, Pythia, Program Your Robot, IRPG, Leek Wars, Gidget, Train B&P, LightBot 2.0, Robot ON!, Prog&Play, Cube Game, The Catacombs, Project Orion, No Bug's Snack Bar, Bomberman Game, Capital Tycoon, Codingame, Code Fights, Saving Sera, Ruby Warrior, EleMental, Screeps, Resource Craft, Critical Mass, Unnamed Prototype, RoboCode, CMX, RoboBUG, Super Mario Collaborative, Code Hunt, Pex4Fun, Soccercode, Program Pacman.

Abaixo seguem exemplos de jogos comerciais feitos para a aprendizagem de lógica de programação:

- Code Combat: jogo open-source desenvolvido por voluntários para plataforma web. Pode ser jogado gratuitamente, porém há opção de comprar assinatura para desbloquear vantagens. Nesse jogo, o jogador deve digitar as instruções na linguagem de programação escolhida. No momento é possível selecionar as linguagens python, javascript, coffeescript e C++. Todas as ações do personagem são guiadas pelo código escrito pelo jogador.



Figura 4. Captura de uma das fases do jogo “Code Combat”.

- Human Resource Machine e 7 Billion Humans: desenvolvidos pela empresa “Tomorrow Corporation”, são jogos pagos. As ações dos personagens são guiadas pelo código elaborado pelo jogador, porém as

instruções são formadas por blocos de comando, e o jogo não utiliza nenhuma linguagem de programação.



Figura 5. Captura de uma das fases do jogo “Human Resource Machine”.



Figura 6. Captura de uma das fases do jogo “7 Billion Humans”.

- Rabbids Coding: jogo gratuito desenvolvido pela empresa Ubisoft. Também utiliza blocos de comando com linguagem natural para guiar as ações do personagem.



Figura 7. Captura de uma das fases do jogo “Rabbids Coding”.

3.3. Efetividade no uso de jogos eletrônicos educativos

Há alguns casos descritos na literatura que discutem a efetividade dos jogos eletrônicos educativos, são exemplos: o estudo de Crocco, Hernandez e Offenholley com jogos educativos em cursos superiores de Matemática, Ciência e Inglês, utilizando métodos quantitativos e qualitativos, concluiu que aspecto de diversão contribuiu para o engajamento e resultou em melhores índices de aprendizado profundo e capacidade de raciocínio lógico.

Clarebout, Vandercruysse e Vandewaetere fizeram uma revisão sistemática da literatura e selecionaram 22 artigos sobre jogos digitais educacionais e observaram que estes estudos apontam os seguintes benefícios: melhor performance; redução de equívocos e erros; redução de stress e frustração; aumento na qualidade de transferência do conhecimento; influência no clima da sala de aula e no comportamento do aluno.

Aranha, Silva e Medeiros também fizeram uma revisão sistemática de estudos realizados sobre jogos digitais para ensino de programação no Brasil e selecionou 17 artigos que mostraram, com exceção de um artigo, que o uso de jogos contribuiu para o rendimento do aluno.

4. DESENVOLVIMENTO

4.1. Ferramentas e componentes

4.1.1. Engine

Conforme os jogos foram evoluindo, mais e mais elementos foram agregados aos projetos e, com a complexidade, o esforço e o volume de trabalho para desenvolver sistemas auxiliares (por exemplo: inteligência artificial, física, renderização, gerenciamento de memória) do jogo se tornaram impraticáveis. Isso porque, devido ao fato de esses sistemas auxiliares poderem ser reutilizados para diferentes projetos, foram criados softwares, ambientes de desenvolvimento, nos quais bibliotecas com essas funções estão disponíveis para utilização, oferecendo ao usuário facilidades que o permitem focar somente na lógica do jogo em si. Esses ambientes são chamados de *Engine*, ou motor de jogo (SANTOS, 2017).

O motor de jogo que foi utilizado neste trabalho se chama Unity, uma *engine* de desenvolvimento multiplataforma 2D e 3D, utilizada primariamente para criação de jogos, mas também utilizada na arquitetura, arte, simulação VR e AR, animações e mais. Unity traz um editor gráfico intuitivo, em que os cenários e os níveis podem ser desenvolvidos dentro de cenas. A programação do jogo aceita pelo Unity deve ser em linguagem C#, utilizando-se o editor Visual Studio. O Unity é uma das *engines* mais populares no mercado, pois, além de ser robusta e de fácil operação, tem versão gratuita e uma grande comunidade que contribui com materiais de aprendizado, como tutoriais, e recursos de jogos (*assets*), sendo apropriada e acessível a desenvolvedores de todos os níveis de conhecimento e experiência (SANTOS, 2017; WILLIAM, 2020).

As versões utilizadas foram:

- Unity 2019.3.15f1
- Visual Studio Community 2019 16.2.5

4.1.2. Assets

De acordo com o site oficial do Unity, assets são coleções exclusivas de recursos para tipos de jogos específicos. Essas coleções podem ser baixadas e importadas no Unity e ficam disponíveis para que o desenvolvedor as utilize em seus jogos.

Os assets utilizados no protótipo desenvolvido estão reunidos no game pack “2D Game Kit”, disponível na Asset Store no site oficial do Unity.

O 2D Game Kit traz elementos prontos, como imagens, áudio, sprites, animações, UIs (menus, info posts, barra de vida etc.), scripts para interação com o personagem jogável, inimigos, objetos e outros. Em resumo, apresenta o necessário para que qualquer pessoa, mesmo sem conhecimento em jogos, possa construir um jogo de aventura 2D.

Na figura 8 é apresentado um resumo dos componentes incluídos no kit.

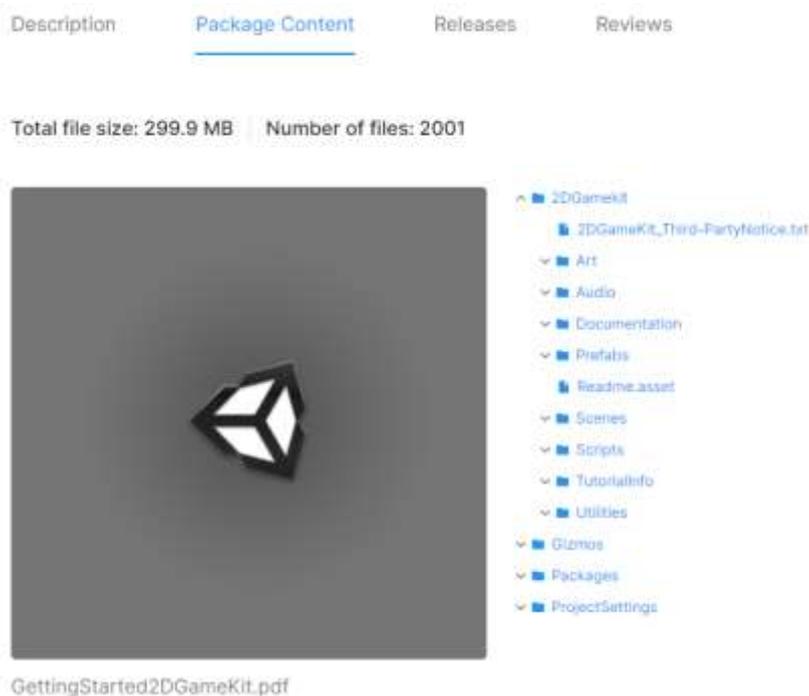


Figura 8. Resumo do conteúdo do “2D Game Kit”

No presente trabalho, os assets foram usados para criar a base do jogo no que se refere ao estabelecimento da narrativa inicial, interação com o jogador e construção do nível similar a um jogo de aventura. A essa base foram feitas modificações e

adições para incorporar à estrutura básica do nível um desafio de lógica de programação.



Figura 9. UI inicial do jogo, parte do “2D Game Kit”.



Figura 10. Exemplo de cenário do “2D Game Kit”.

4.1.3. Ferramentas gráficas

Foram utilizados 2 softwares para confecção de imagens utilizadas no protótipo:

- a. Gimp, para criação do conteúdo teórico que compõe parte do elemento UI “diário” da personagem. Gimp é um editor de imagem multiplataforma gratuito e open-source.
- b. Draw.io, para criação do diagrama de Chapin que compõe parte do elemento UI “terminal” da personagem. É um software de diagramação gratuito e open-source.

As versões utilizadas foram:

- Gimp 2.10.18
- Draw.io 13.0.3

4.2. Metodologia

Os recursos utilizados para guiar o desenvolvimento deste protótipo de jogo foram coletados por meio de pesquisa na ferramenta de busca “Google Acadêmico”. Foram buscados artigos, livros, palestras e trabalhos acadêmicos, num período de 2000 a 2020.

A estrutura básica do protótipo foi baseada principalmente na estrutura descrita no livro “Rules of Play”, de Salen e Zimmerman.

O foco do design de jogo é a estrutura através da qual os jogadores serão guiados em suas aventuras. As regras devem definir essa estrutura e criar as experiências idealizadas, devendo ser intuitivas e não restritas demais, pois o jogador deve ser capaz de se divertir com os desafios sem se sentir frustrado (SALEN & ZIMMERMAN, 2004).

De acordo com Salen e Zimmerman (2004), regras são as restrições formais no desenho do jogo. São construções lógicas e matemáticas. Jogabilidade são ações inseridas no jogo propositalmente para possibilitar a participação do jogador com o jogo e os outros jogadores. Cultura são aspectos do jogo que refletem o contexto cultural do personagem/mundo criado. Salen e Zimmerman ainda salientam que os três elementos estruturais fundamentais não são mutuamente exclusivos, pois um elemento pode ser a combinação dele mesmo com outro.

Costa (2014) também faz uma conexão entre esta estrutura e o contexto educacional: Regras: é o que pautará a quantidade de conteúdo que poderá ser acessado, assim como definir como será medido o desempenho do aluno; Jogabilidade: o conhecimento será compartilhado através dos desafios e/ou narrativa do jogo; Cultura: fará a conexão entre o conhecimento adquirido e o mundo real, passando a confiança necessária para que o aluno possa obter domínio do assunto

Para realizar a construção dos cenários e scripts, foram utilizados tutoriais do site oficial do *Unity Learn* e consultas ao site *stackoverflow*.

O conteúdo pedagógico foi baseado no livro de Almeida “Curso essencial de lógica de Programação”, principalmente na construção do pseudocódigo e diagrama. Já a abordagem pedagógica baseia-se na teoria construtivista de Ausubel, a aprendizagem significativa.

Para elaborar o GDD (Game Design Document) adicionado ao apêndice deste trabalho, foi utilizado como base a estrutura de GDD para jogos educativos descrita por Leite e Mendonça (2013).

4.3. Estrutura do Protótipo

O protótipo de jogo foi desenvolvido no presente trabalho para apresentar um conceito de jogo de aventura intercalado com desafios de lógica, com os seguintes objetivos: possibilitar a aplicação da prática dos conceitos de lógica de programação e promover aspectos lúdicos e motivacionais para engajamento do jogador.

Uma metodologia pedagógica para esse fim que foi citada por alguns autores na literatura revisada é a aprendizagem significativa. Novos conhecimentos podem sempre ser mais facilmente adquiridos de forma natural, sem memorização, ao entendermos realmente o seu significado e como os diferentes conceitos se conectam. Para que esse objetivo fosse alcançado, o protótipo foi desenhado de forma que o jogador pudesse influenciar a história e o andamento do jogo através da modificação dos algoritmos dentro do desafio de lógica – ou seja, através da interação e da tentativa e erro o aluno pode inferir no significado dos conceitos estudados. Para que fosse atingido o propósito lúdico e de engajamento do aluno, foi proposta a separação entre a codificação e as ações do personagem, nas quais, em outros jogos,

cada ação do personagem deve ser primeiramente escrita em forma de comando, evitando que o jogo se torne monótono. Buscou-se incorporar ao jogo os seguintes elementos, conforme indicados por Costa (2014): puzzle, renovação, alvo, julgamento, dado e reconhecimento, através do desafio de lógica apresentado no jogo. E os seguintes elementos motivacionais, conforme indicou Peres (2016): fluxo e imersão através da jogabilidade, que permitem ao jogador a concentração em busca das chaves e de derrotar os inimigos e, depois, concentrar-se no desafio, fazendo essa separação com o objetivo de manter o estado de fluxo e de imersão.

4.3.1. Jogabilidade e regras

Parte importante da jogabilidade é a narrativa que direciona o jogador. Neste protótipo, a personagem principal é uma engenheira intergaláctica e seu trabalho é viajar pelo universo à procura de aventuras e dispositivos para reparar. A protagonista encontra o planeta “90093”, gerenciado por sistemas de computador que foi abandonado por causa de suas muitas falhas de programação, o que o tornou inabitável.



Figura 11. Introdução da história por meio de infopost.

O jogador é desafiado, então, a tornar o planeta funcional novamente, propondo que a protagonista acesse o código do sistema principal e conserte as partes dos

algoritmos que apresentam falha de lógica. Com isso, é possível introduzir ao jogo desafios de lógica para prática de lógica de programação, que não só tem impacto direto na progressão do jogo, como configura o desafio principal a ser alcançado, fazendo com que o jogador se mantenha motivado e disposto a analisar o problema.

Embora os desafios de lógica sejam o objetivo principal da proposta do jogo, ele deve ser alcançado somente após o personagem encontrar as chaves de segurança (*tokens*) que desbloqueiam o código para edição. O personagem deve percorrer o cenário, evitar perigos e derrotar inimigos para conseguir obter estas chaves. Ao utilizá-las no terminal remoto (um item que a protagonista usa para acessar o sistema do planeta), o jogador pode alterar e corrigir a lógica, vencendo o desafio de lógica proposto.



Figura 12. Protagonista encontra uma chave de segurança.

A personagem principal possui um diário onde faz anotações que ajudam o jogador a identificar o erro no algoritmo apresentado e traz, na aba, teoria breve, com definições de elementos usados na construção de algoritmos.

Diário

Diagramas | Teoria

Portal 001

VAR
Portal, Planeta : inteiro
Ativar_portal : booleano

Neste diário encontram-se diagramas com os códigos corretos, entre na aba 'Teoria' para entender melhor as anotações.

Clique em Terminal para ir ver e corrigir o código.

Terminal < >

Figura 13. Diário pertencente à protagonista, aba Diagrama.

Diário

Diagramas | Teoria

Algoritmos

Um algoritmo consiste em uma série limitada de etapas, cujo objetivo é executar uma determinada tarefa. Melhor dizendo, é uma série de instruções ordenadas de maneira lógica para solucionar determinado problema. Para criar um algoritmo, é preciso dividir o problema em questões em três etapas importantes: entrada, processamento e saída.

Exemplo de algoritmo: Ler dois valores (N1 e N2) [Entrada], somar os dois valores [Operação], e mostrar o resultado da soma [Saída].

```

VAR N1, N2
INÍCIO
  Leia N1;
  Leia N2;
  Soma = N1 + N2;
  Escreva Soma;
FIM
  
```

< >

Figura 14. Diário pertencente à protagonista, aba Teoria.

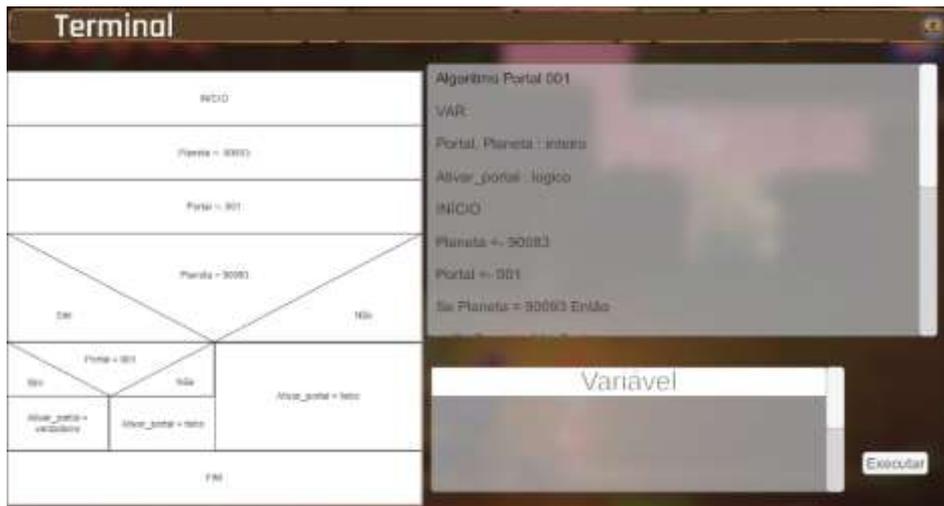


Figura 15. Terminal pertencente à protagonista.

As chaves ficam disponíveis até que o jogador acerte a sessão do código que diz respeito à cada chave. Para habilitar o código para edição, o jogador deve clicar na chave, fazer as modificações que quiser e apertar o botão “Executar” para que seja feita a verificação. São quatro chaves, cada uma desbloqueia uma sessão do algoritmo: variável, operação, condicional e repetição.

- Variável habilita a seção entre VAR e INÍCIO.
- Condicional habilita todo o bloco dentro de uma estrutura condicional.
- Repetição habilita todo o bloco dentro de uma estrutura de repetição.
- Operação habilita todas as outras instruções fora das três seções definidas acima.

4.3.2. Cultura: representação de algoritmo

No presente trabalho, foram utilizados dois tipos de representações de algoritmo dentro do protótipo do jogo: a. o Diagrama Nassi-Shneiderman ou diagrama de Chapin, e b. pseudocódigo.

O Diagrama de Nassi-Shneiderman ou diagrama de Chapin é uma representação gráfica de um algoritmo criada por Ned Chapin, baseado fortemente nas obras de Nassi e Shneiderman, em 1973. Utiliza-se de quadros organizados de forma hierárquica para representar a sequência ordenada de instruções lógicas.

Originalmente criado para melhor representar programação estruturada e iterativa, esse modelo era considerado superior pelos seus criadores em termos de definição e clareza, especialmente de estruturas encapsuladas, como acontece com estruturas condicionais e de repetição, principalmente em comparação com o modelo de fluxograma da época (BARCELOS, 2012; NASSI & SHNEIDERMAN, 1973).

Almeida define que “o intuito deste diagrama é representar os comandos de um algoritmo dentro de um único retângulo, subdividindo-o em retângulos menores, que representam os diferentes blocos de sequência de comando do algoritmo”.

Retângulos representam comandos, e triângulos representam uma condição a ser testada logicamente. Como por exemplo, na Figura 16.:



Figura 16. Diagrama de Chapin. Retirado do livro “Curso essencial em lógica de Programação”, 2008.

O Pseudocódigo, ou linguagem estruturada, usa linguagem natural para descrever em detalhes o algoritmo. Sua estrutura incorpora os elementos de construção do algoritmo, como condicionais, por exemplo, e por isso se parece muito com o código em linguagem de programação. Sem padrão único, sua função é compor a lógica do algoritmo sem se preocupar com regras de sintaxe (ALMEIDA, 2008; BARCELOS, 2012).

4.3.3. Conceitos abordados

Neste protótipo foi abordado conceitos de variável. O desafio apresentado ao jogador testa o conhecimento do aluno sobre atribuição de variável.

Segundo Almeida (2008), variável pode ser definida como “Variável é a representação simbólica dos elementos de certo conjunto. Cada variável ocupa uma posição de memória, cujo conteúdo pode se modificar durante a execução de um programa.”

Sobre o comando de atribuição, Almeida afirma que:

- O tipo de dado atribuído deve ser compatível com o tipo declarado.
- Geralmente representado por uma seta.
- Sintaxe básica é: identificador <- expressão. Onde o identificador é o nome da variável que irá receber o valor resultado da expressão.

4.4. Protótipo do jogo “The Explorer”

4.4.1. Nível

Foi desenvolvido um nível para demonstração do protótipo. Nele, há duas cenas. A cena 1 apresenta a personagem, e a segunda introduz a história e o desafio a ser conquistado.



Figura 17. Cena 1.

O objetivo na segunda cena é desbloquear a entrada do portal principal, e, para isso, o desafio é corrigir o erro no algoritmo que habilita a passagem.



Figura 18. Cena 2.

No protótipo apresentado, é representado um desafio de dificuldade “fácil” e por isso o jogador pode ver no diário que a protagonista desenhou um diagrama com o algoritmo corrigido. A ideia é familiarizar o jogador com a interpretação e análise dos diagramas como forma de representação de algoritmos e, ao mesmo tempo, poder

compará-lo com um algoritmo em pseudocódigo. O jogador poderá fazer essa comparação diretamente no “Terminal”.

O algoritmo apresentado não é difícil e tem como objetivo estabelecer a mecânica do jogo e familiarizar o jogador com os elementos de programação apresentados, que neste caso são: atribuição de variáveis e tipos de dados.

O nível possui duas chaves: variável e operação. O jogador só precisará da chave operação para corrigir a atribuição feita na variável Planeta, que está incorreta no código: 90083.

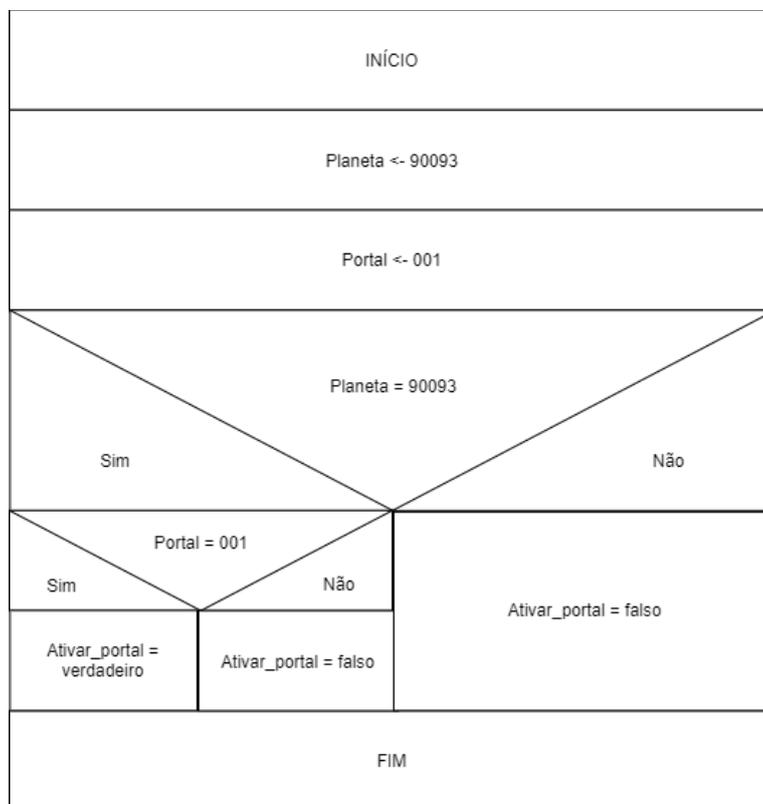


Figura 19. Diagrama com o algoritmo corrigido, desafio 1.

```

1 Algoritmo Portal 001
2 VAR
3 Portal, Planeta : inteiro
4 Ativar_portal : logico
5 INÍCIO
6 Planeta <- 90083
7 Portal <- 001
8 Se Planeta = 90093 Então
9     Se Portal = 001 Então
10        Ativar_portal <- verdadeiro
11        Senão
12            Ativar_portal <- falso
13        Fimse
14    Senão
15        Ativar_portal <- falso
16    Fimse
17 FIM

```

Figura 20. Código com falha, desafio 1.

4.4.2. Scripts

Para o desenvolvimento da UI Diário e Terminal, foram criados scripts para navegação e verificação do algoritmo. Também foram feitas modificações no sistema de “chaves” que veio pronto do game pack.

UI Diário: composto por DiaryCanvas e subsequentes game objects.

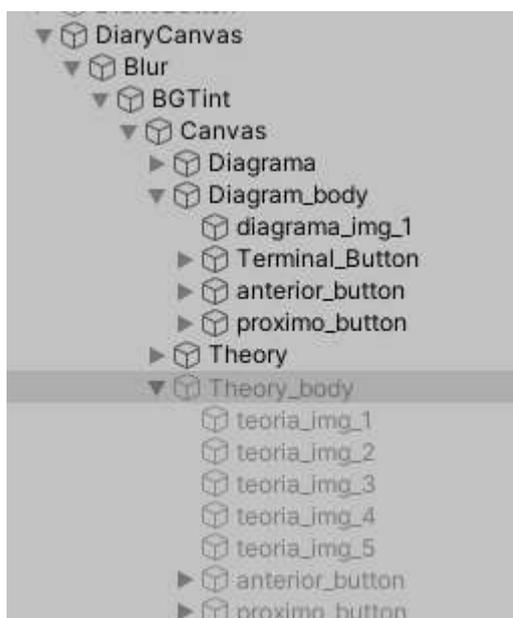


Figura 21. Visão hierárquica da UI Diário

DiaryNav.cs: script para navegação entre as páginas do diário. Adicionado aos objetos Theory_body e Diagram_body.

```
6 public class DiaryNav : MonoBehaviour
7 {
8     public GameObject atual;
9     public GameObject[] DList;
10
11     O referências
12     public void Prox()
13     {
14         int a = Array.IndexOf(DList, atual);
15         GameObject prox = DList[a + 1];
16         if(prox != null)
17         {
18             atual.SetActive(false);
19             atual = prox;
20             atual.SetActive(true);
21         }
22     }
23 }
```

Figura 22. Excerto do script DiaryNav.cs

UI Terminal: composto por TerminalCanvas e subsequentes game objects.

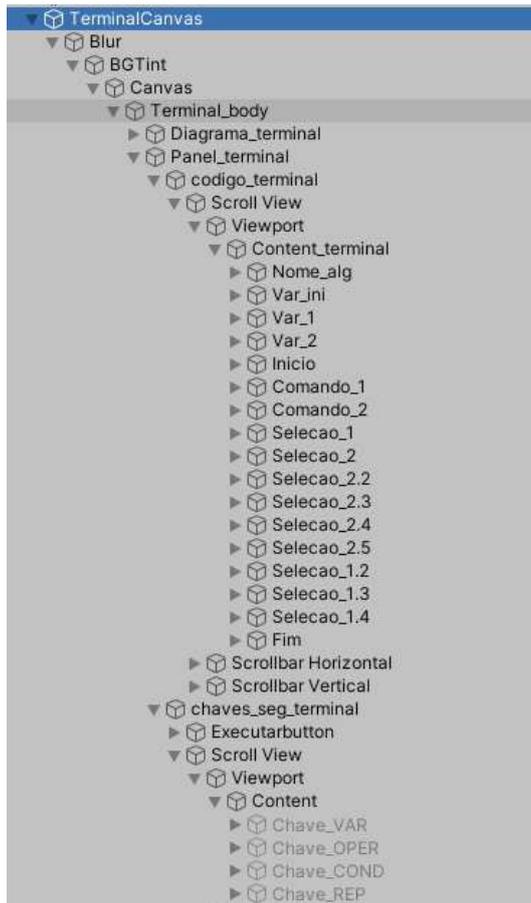


Figura 23. Visão hierárquica da UI Terminal

KeyCodeEnable.cs: habilita para edição os InputFields com a tag submetida. Adicionado ao objeto Content_terminal, sua função EnableCoding é chamada pelos objetos Chave.

Exemplo: Ao clicar na chave “variável” (game object “Chave_VAR”), serão habilitados todos os InputFields com a tag “VAR”.

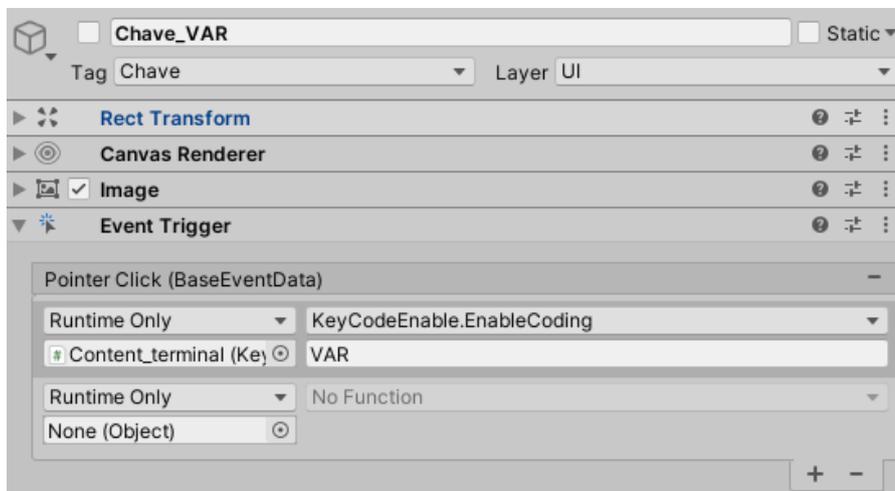


Figura 24. Inspector do objeto Chave_VAR

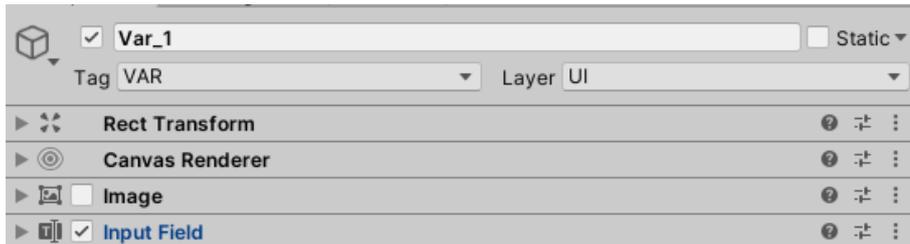


Figura 25. Inspector do objeto Var_1

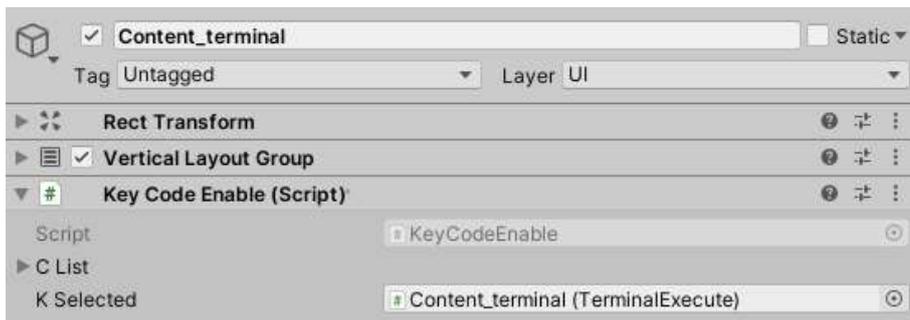


Figura 26. Inspector do Content_terminal com KeyCodeEnable.

```

11 public void EnableCoding(string tipo)
12 {
13     List<string> selCode = KSelected.SelectedCodeList;
14
15     // Desable all input fields
16     InputField[] alllines = GameObject.Find("Content_terminal").GetComponentsInChildren<InputField>();
17     foreach (InputField al in alllines)
18     {
19         al.interactable = false;
20     }
21
22     // Select only fields of key type selected
23     CList = GameObject.FindGameObjectsWithTag(tipo);
24     if (!selCode.Contains(tipo))
25     {
26         selCode.Add(tipo);
27     }
28
29     // Enable fields of key type selected
30     if(CList != null)
31     {
32         foreach (GameObject codeline in CList)

```

Figura 27. Excerto do script KeyCodeEnable.cs

TerminalExecute.cs: Faz a checagem do código alterado no terminal. Se correto, o game object que faz a transição de cena será habilitado. Esse código é específico para este nível, por conter hard coding.

Adicionado ao objeto Content_terminal, sua função Execute é chamada pelo objeto Executarbutton.

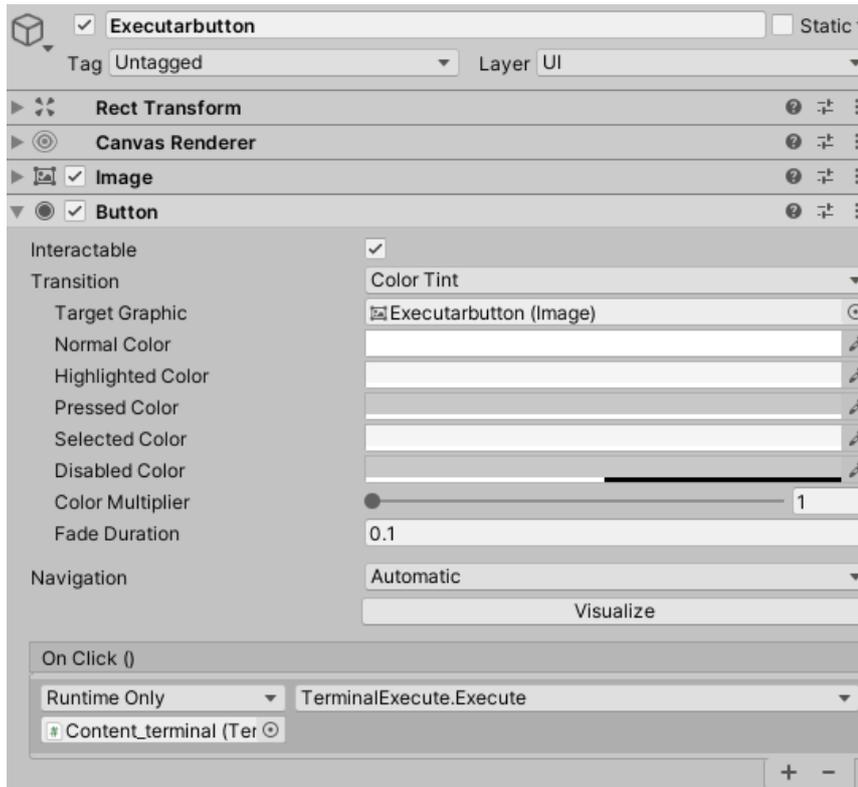


Figura 28. Inspector do objeto Executarbutton

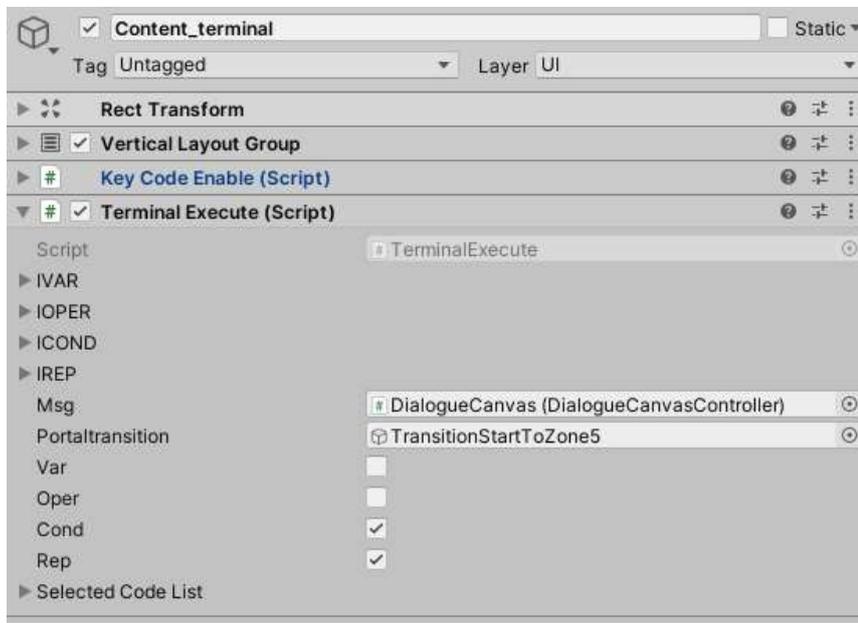


Figura 29. Inspector do Content_terminal com TerminalExecute

```

55 public void Execute()
56 {
57     foreach(string item in SelectedCodeList)
58     {
59         switch (item)
60         {
61             case "VAR":
62                 CheckVAR();
63                 break;
64             case "OPER":
65                 CheckOPER();
66                 break;
67             case "COND":
68                 CheckCOND();
69                 break;
70             case "REP":
71                 CheckREP();
72                 break;
73             default:
74                 break;
75         }
76     }

```

Figura 30. Excerto do script TerminalExecute.cs

Uma modificação foi feita no padrão que veio com o game pack quando a personagem coletar uma chave, no evento OnExit(): o objeto Chave_xxx é habilitado para ser usado no terminal.

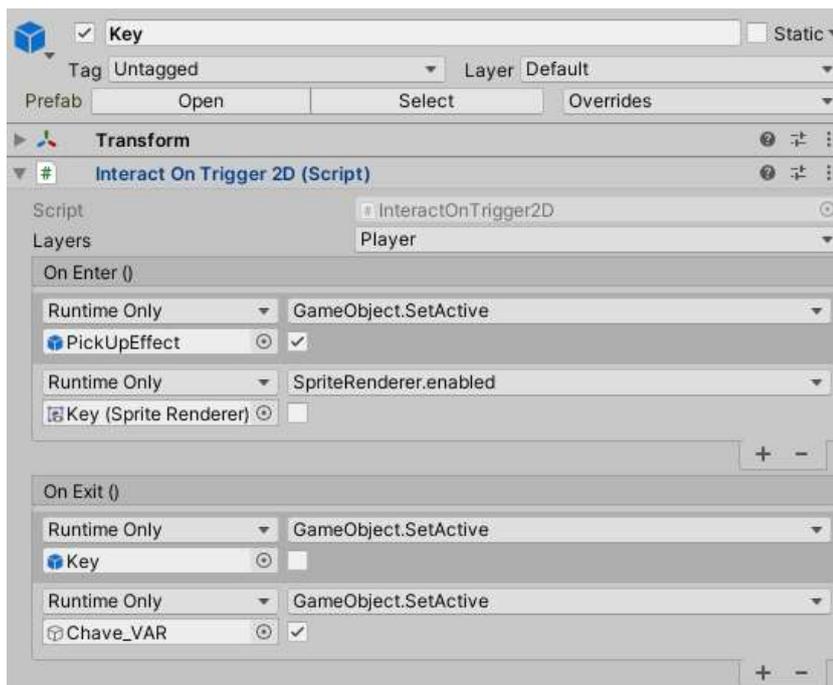


Figura 31. Inspector do game object Key

5. CONCLUSÃO

Muito pode ser alcançado com o auxílio da tecnologia. O uso dos jogos eletrônicos na educação traz grandes vantagens, e mesmo suas desvantagens podem ser evitadas se eles forem utilizados com inteligência. É, por suposto, naturalmente, que os jogos são uma ferramenta disponível a serviço do educador e este deve ser sempre presente como mentor e mediador entre o conhecimento e a compreensão do conteúdo e a atividade proposta pelo jogo.

Não é difícil de entender por que os jogos são ótimas ferramentas de aprendizado, já que o ato de jogar é tão apreciado fora da sala de aula. Isso se dá por suas características básicas, seja em jogos educacionais ou de entretenimento: desafios que engajam o jogador e os motiva a continuar tentando apesar dos erros, possuem formato dinâmico, interativo e visual, o que evita que o aluno sinta tédio e monotonia, além de facilitar a demonstração do conteúdo através de recursos como a simulação.

As pesquisas feitas até o momento mostram que o uso dos jogos educativos promove grandes benefícios, porém maiores pesquisas são necessárias para entender como aspectos específicos dos jogos influenciam os processos de ensino-aprendizagem. Para isso, também é necessário entender como as metodologias pedagógicas melhor se relacionam com os conteúdos a serem trabalhados e que aspectos dos jogos podem oferecer vantagens sobre as formas de ensino tradicionais.

Com isso em mente, este trabalho se propôs a desenvolver um protótipo de jogo 2D para a prática de lógica de programação. No que diz respeito aos elementos de jogo incorporados ao protótipo, os objetivos previstos para a fase de desenvolvimento em que se encontra o protótipo foram atingidos, possibilitando que seja incorporada prática de conceitos de programação ao jogo, de ludicidade e engajamento ao permitir que o jogador tenha a experiência de jogar um jogo de aventura sem que seja exigida a entrada de comandos de programação o tempo todo.

Ainda há muitos pontos de melhoria futura no protótipo apresentado, tais como: otimização do código no script do terminal, principalmente a eliminação de codificação rígida; ampliação das fases, com mais desafios e diferentes níveis de dificuldade; revisão da metodologia pedagógica por profissional da educação; conclusão de uma versão completa e subsequente estudo de caso e avaliação do jogo.

REFERÊNCIAS

ALMEIDA, Marilane. **Curso essencial de lógica de Programação**. São Paulo: Digerati Books, 2008.

ARANHA, Eduardo Henrique da Silva; SILVA, Thiago Reis da; MEDEIROS, Tainá Jesus. Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. **CINTED-UFRGS: Novas Tecnologias na Educação**. V. 11, Nº 3, dezembro/2013.

Disponível em: https://www.researchgate.net/publication/281375371_Ensino_de_programacao_utilizando_jogos_digitais_uma_revisao_sistemica_da_literatura
Acesso em Dez/2020.

BARBOSA, Leônidas da Silva. **Aprendizado significativo aplicado ao ensino de algoritmos**. Dissertação (Mestrado em Sistemas e Computação). Universidade Federal do Rio Grande do Norte. Natal, 2011. Disponível em: https://repositorio.ufrn.br/jspui/bitstream/123456789/18018/1/LeonidasSB_DISSERT.pdf. Acesso em Jan/2020.

BARCELOS, Ricardo José dos Santos. **O processo de construção do conhecimento de algoritmos com o uso de dispositivos móveis considerando estilos preferenciais de aprendizagem**. Tese (Doutorado em Informática em Educação) Universidade Federal do Rio Grande do Sul. Porto Alegre, 2012.

Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/80524/000904063.pdf?sequence=1>. Acesso em Jan/2020.

BEDER, Delano Medeiros; OTSUKA, Joice Lee. **Jogos educacionais abertos**. In: VALENTE, José Armando; FREIRE, Fernanda Maria Pereira; ARANTES, Flávia Linhalis. – **Tecnologia e educação: passado, presente e o que está por vir**. Campinas, SP: NIED/UNICAMP, 2018. Disponível em:

<https://www.nied.unicamp.br/wp-content/uploads/2018/11/Livro-NIED-2018-final.pdf>. Acesso em Jan/2020.

BIELSCHOWSKY, Carlos Eduardo. Tecnologia da informação e comunicação das escolas públicas brasileiras: o programa proinfo integrado. **Revista e-curriculum**. São Paulo. Vol 5, nº1. Dezembro de 2009. Disponível em:

<https://revistas.pucsp.br/index.php/curriculum/article/view/3256/2174>. Acesso em Jan/2020.

CETIC. **Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras: TIC educação 2017**. São Paulo: Comitê Gestor da Internet no Brasil, 2018. Disponível em: <https://cetic.br/publicacao/pesquisa-sobre-o-uso-das-tecnologias-de-informacao-e-comunicacao-nas-escolas-brasileiras-tic-educacao-2017/>. Acesso em Jan/2020.

CLAREBOUT, Geraldine; VANDERCRUYSSSE, Sylke; VANDEWAETERE, Mieke. Game based learning: A review on the effectiveness of educational games. In CRUZ-CUNHA, Maria Manuela. **Handbook of Research on Serious Games as**

Educational, Business, and Research Tools. USA: IGI Global, 2012. Cap. 32, pág. 628-647. Disponível em: https://www.researchgate.net/publication/260360868_Game-Based_Learning_A_Review_on_the_Effectiveness_of_Educational_Games. Acesso em Dez/2020.

CLUA Esteban Walter Gonzalez; BITTENCOURT, João Ricardo. Desenvolvimento de Jogos 3D: Concepção, Design e Programação. *In: XXV Congresso da Sociedade Brasileira de Computação.* 2005. Disponível em: <http://www2.ic.uff.br/~esteban/files/Desenvolvimento%20de%20jogos%203D.pdf>. Acesso em Jan/2020.

COSTA, Estevan Braz Brandt. **Ambiente de aprendizado de estrutura de dados usando gamificação.** Dissertação (Mestrado em Ciência da Computação). Universidade Estadual de Londrina. Londrina, 2014. Disponível em: <http://www.bibliotecadigital.uel.br/document/?code=vtls000200617>. Acesso em Jan/2020

CROCCO, Francesco; OFFENHOLLEY, Kathleen; HERNANDEZ, Carlos. A Proof-of-Concept Study of Game-Based Learning in Higher Education. **Simulation & Gaming.** USA: SAGE, 2016. Vol 47. Pág. 403–422. Disponível em: https://www.researchgate.net/publication/305041383_A_Proof-of-Concept_Study_of_Game-Based_Learning_in_Higher_Education. Acesso em: Dez/2020.

HAGUENAUER, Cristina Jasbinscheck et al. Uso de Jogos na Educação Online: a Experiência do LATEC/UFRJ. **Revista Educa Online.** Rio de Janeiro. Vol 1, nº1. Janeiro/abril de 2007. Disponível em: <http://files.faculdadede.webnode.com.br/200000029-1bf3c1de7c/Usode%20Jogosna%20Educa%C3%A7%C3%A3o%20Online%20-%20a%20Experi%C3%Aancia%20do%20LATEC-UFRJ.pdf>. Acesso em Jan/2020.

HERNÁNDEZ-RAMOS, P.F.; BACHEN, C. M.; RAPHAEL, C.; WALDRON, A. How do presence, flow, and character identification affect players' empathy and interest in learning from a serious computer game? **Computers in Human Behavior**, junho de 2016. Disponível em: https://www.researchgate.net/publication/305505232_How_do_presence_flow_and_character_identification_affect_players%27_empathy_and_interest_in_learning_from_a_serious_computer_game. Acesso em Jan/2020.

HORTA, Alberto Sire; SILVA, Rafael Araújo. **Desenvolvimento de jogos educacionais utilizando ambientes virtuais de aprendizagem através de role-playing games.** Monografia (Graduação em Computação Científica). Universidade de Taubaté. Taubaté, SP. 2011.

HUIZINGA, Johan. **Homo Ludens: o jogo como elemento da cultura.** 4ª ed. São Paulo, SP: Perspectiva, 2000.
IBRAHIM, Roslina; YUSOFF, Rasimah Che Mohd; MOHAMED, Hasiah; JAAFAR, Azizah. Students Perceptions of Using Educational Games to Learn Introductory

Programming. **Canadian Center of Science and Education**.p. 205-216. Vol. 4, No. 1. January 2011.

KISHIMOTO, André. **Inteligência Artificial em Jogos Eletrônicos**. Universidade Presbiteriana Mackenzie. São Paulo, 2004. Disponível em: <http://www.kishimoto.com.br/publications.php>. Acesso em Jan/2020

LEITE, Patricia da Silva; MENDONÇA, Vinícius Godoy de. Diretrizes para Game Design de Jogos Educacionais. *In: XII Simpósio Brasileiro de Jogos e Entretenimento Digital. Art & Design Track – Full Papers*. 2013. Disponível em: <https://www.sbgames.org/sbgames2013/proceedings/artedesign/17-dt-paper.pdf>. Acesso em: Dezembro/2020.

MEGACURIOSO. Senet, o jogo para falar com os mortos no egito antigo. **No Zebra Network**. Fevereiro de 2020. Disponível em: <https://www.megacurioso.com.br/artes-cultura/113465-senet-o-jogo-para-falar-com-os-mortos-no-egito-antigo.htm>. Acesso em Fev/2020.

MICHAELIS. **Dicionário Brasileiro da Língua Portuguesa**. Melhoramentos, 2020. Disponível em: <https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/algorithm/>. Acesso em Jan/2020.

MILJANOVIC, Michael A.; BRADBURY, Jeremy S. **A Review of Serious Games for Programming** University of Ontario Institute of Technology, 2000. Disponível em: https://www.researchgate.net/publication/327663695_A_Review_of_Serious_Games_for_Programming. Acesso em Outubro/2020.

MORATORI, Patrick Barbosa. **Por que utilizar jogos educativos no processo de ensino aprendizagem?**. Dissertação (Mestrado de Informática Aplicada à Educação). Universidade Federal Do Rio De Janeiro. Rio De Janeiro, 2003. Disponível em: http://www.nce.ufrj.br/ensino/posgraduacao/strictosensu/ginape/publicacoes/trabalhos/t_2003/t_2003_patrick_barbosa_moratori.pdf. Acesso em Outubro/2020.

NASSI, I ; SHNEIDERMAN, B. **Flowchart Techniques for structured programming**. SIGPLAN Notices. State University of New York, Agosto de 1973. Disponível em: <http://www.cs.umd.edu/~ben/papers/Nassi1973Flowchart.pdf>. Acesso em Jan/2020.

NUNES, Mariana Monteiro. **Desenvolvimento de um ambiente virtual de treinamento colaborativo para a disciplina de estrutura de dados**. Dissertação (Mestrado em Ciências da Computação). Universidade Estadual de Londrina. Londrina, 2016.

OUTERSPACE. **A História dos Videogames**. Disponível em: <https://www.outerspace.com.br/retrospace/materias/consoles/historiadosconsoles1.htm>. Acesso em Fev/2020.

PERES, Frederico Oldemburgo. **Análise da correlação entre os elementos motivacionais e o sucesso de jogos sociais do tipo bolha**. Dissertação (Mestrado em Ciências da Computação). Universidade Estadual de Londrina.

Londrina, 2016. Disponível em:
<http://www.bibliotecadigital.uel.br/document/?code=vtls000208813>. Acesso em Jan/2020.

PHYE, Gary D. **Handbook of academic learning**. San Diego, CA: Academic Press, 1997.

RAPKIEWICZ, Clevi Elena et al. Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. *In: CINTED-UFRGS Novas Tecnologias na Educação*. Dezembro de 2006. Disponível em:
<https://seer.ufrgs.br/renote/article/view/14284>. Acesso em Jan/2020.

ROCHA, Rafaela Vilela da; BITTENCOURT, Ig Ibert; ISOTANI, Seiji. Análise, Projeto, Desenvolvimento e Avaliação de Jogos Sérios e Afins: uma revisão de desafios e oportunidades. *In: Anais do XXVI Simpósio Brasileiro de Informática na Educação*. 2015. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/5342>. Acesso em: Jan/2020.

SAID, Ricardo. **Curso de Lógica de Programação**. São Paulo: Digerati Books, 2007.

SALEN, Katie; ZIMMERMAN, Eric. **Rules of play: game design fundamentals**. Cambridge, MA: The MIT Press, 2004.

SANTOS, Flávia Moreira dos. **Aprenda Unity: Criando Jogo de Nave 2D**. 2017.

SAKUDA, Luiz Ojima.; FORTIM, Ivelise; ROLIM, Amanda, JANILLE, Erica. Perfil da Indústria Brasileira de Jogos Digitais (IBJD). *In: SAKUDA, Luiz Ojima; FORTIM, Ivelise (Orgs.). II Censo da Indústria Brasileira de Jogos Digitais*. Ministério da Cultura: Brasília, 2018.

SILVA, Signe Melo et al. **Design Instrucional para uma Aprendizagem Significativa: pesquisa e extensão no caminho do desenvolvimento de tecnologias para a educação**. *In: Congresso Regional sobre Tecnologias na Educação*, 2016. Disponível em: http://ceur-ws.org/Vol-1667/CtrIE_2016_AC_paper_57.pdf. Acesso em Jan/2020.

SOUSA, Marcone Hilton de. Meaningful Game: um olhar sobre o uso de jogos e aprendizagem significativa na educação. *In: XII Seminário SJECC*. 2017. Disponível em: <https://www.revistas.uneb.br/index.php/sjec/article/view/3555>. Acesso em Jan/2020.

WERNER, Claudete; GONÇALVES, Sergio Henrique Torres. Conceitos e Desenvolvimento de Jogos Digitais Educativos. **Anais da XIII Semana de Informática e XV Mostra de Trabalhos de Iniciação Científica**. Paranavaí, PR. 2016.

WILLIAM, Emma. **Learning C# by developing games with Unity**. Julho, 2020.

APÊNDICE A – Game Design Document

The Explorer

Resumo do jogo

The Explorer é um protótipo de jogo educativo para prática de lógica de programação. Ellen, a personagem principal, é uma exploradora intergaláctica com a missão de explorar um mundo misterioso e sem habitantes (exceto monstros inimigos).

História do jogo

Ellen descobre que pode reabilitar o planeta se corrigir os erros no sistema de gerenciamento do planeta. A protagonista deve lutar contra inimigos, sobreviver a perigos do ambiente e coletar chaves para que possa habilitar a edição do código corrompido. O desafio final é corrigir a lógica do código para restaurar o funcionamento normal do planeta.

Objetos essenciais:

Armas: pistola laser (munição infinita); bastão (inquebrável) – Atinge objetos quebráveis e inimigos, dano causado: 1 unidade.

Objetos: chaves de segurança - habilita edição de seção de código específica no terminal.

Inimigos: spitfire (5 unidades de vida) – dano a longa distância: 1 unidade; espinhos – dano na colisão: 1 unidade.

Características

Perspectiva: narrativa linear, em terceira pessoa.

Gênero: Aventura; puzzle; ambientada em um mundo alienígena.

Plataformas: Windows.

Quantidade de níveis: 1 nível

Quantidade de vilões/inimigos: 1 inimigo

Quantidade de jogadores: 1 jogador

Público-alvo: Estudantes de lógica de programação

Puzzles: Pseudocódigo com erro de lógica, deve ser corrigido pelo jogador.

Interface/Controles

Pular – Espaço

A – Andar para esquerda

W – Olhar para cima

S – Abaixar

D – Andar para direita

O – Atirar

K – Bater com bastão

E – Botão de ação

Desafio

O protótipo traz um desafio de nível fácil.

Contém 1 inimigo que causa dano a longa distância.

Contém 1 elemento de cenário que causa dano ao haver colisão com o personagem.

Vidas do personagem: 5 unidades.

O desafio de lógica pode ser encontrado no diário da personagem. Objetivo é corrigir um erro de atribuição de variável no pseudocódigo apresentado no terminal.

Objetivos de aprendizagem: Prática do conceito de variável e atribuição de variável. Interpretação de diagrama de Chapin e pseudocódigo.

Avaliação do aluno: Auto-avaliação baseada no feedback provido pelo jogo. Ao executar o código, o jogador recebe feedback imediato informando o acerto ou erro.