

# As Aventuras Espaciais de Cody: protótipo de jogo para auxiliar no ensino de lógica de programação

Ian Macedo Maiwald Santos<sup>\*</sup> Jhemeson Silva Mota Alexandre Henrique Kavalerski Teixeira  
Jackson Gomes de Souza

Centro Universitário Luterano de Palmas, Departamento de Computação, Brasil

## RESUMO

O constante surgimento e a evolução das ferramentas computacionais expandem a gama de possíveis soluções inovadoras resultantes da associação da computação com as mais diversas áreas de conhecimento. Dentre as áreas que têm grande intersecção com a computação, pode-se citar a educação, principalmente se associada a elementos de jogos. O presente trabalho tem o objetivo de desenvolver um protótipo de um jogo sério onde o jogador, para realizar as devidas ações deverá dar comandos básicos que serão submetidos a um processo de tradução e resultarão nos movimentos do personagem. Desta forma, o jogador aprenderá lógica de programação enquanto se diverte.

**Palavras-chave:** programação, ensino de programação, compilador.

## 1 INTRODUÇÃO

A integração entre as novas tecnologias da informação e a educação resulta em ferramentas com recursos de criação e propagação de conhecimento que facilitam o processo de ensino-aprendizagem [1]. O uso destas ferramentas agregadas às habilidades técnicas e pedagógicas dos professores pode permitir reestruturar a relação professor-aluno e potencializar estratégias didáticas em sala de aula [2].

O ensino de lógica de programação se mostra eficaz para estimular o desenvolvimento de habilidades como resolução de problemas e criatividade [3]. Os jogos educativos destacam-se dentre as ferramentas capazes de auxiliar na aprendizagem por possuírem estruturas semelhantes à situação de aprendizado escolar e proverem uma experiência lúdica que possibilita que o jogador aprenda mesmo sem perceber que está sendo ensinado [4].

O presente trabalho apresenta o jogo “As Aventuras Espaciais de Cody” que tem como objetivo, ensinar e treinar noções iniciais de lógica de programação. Ao longo das próximas seções, serão abordados os conceitos envolvidos na produção do jogo, as ferramentas e métodos envolvidos no trabalho e o protótipo desenvolvido.

## 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os conceitos envolvidos no presente trabalho, bem como a fundamentação teórica de cada um deles. A subseção 2.1 aborda conceitos, características e elementos de jogos. Já a seção 2.2 trata do ensino de lógica de programação. A subseção 2.3 apresenta o funcionamento das diversas etapas envolvidas no processo de compilação.

### 2.1 Jogos

Jogos são elementos primitivos que estão presentes não somente na vida humana, mas na vivência de diversos animais e precedem o surgimento da cultura, além disso, são atividades sociais que têm como características o prazer, a liberdade, a separação dos fenômenos do cotidiano, as regras, o caráter fictício ou representativo, sua limitação no tempo e no espaço e o caráter “não-sério” [5]. Um jogo pode ser entendido como um sistema onde jogadores envolvem-se em um conflito artificial definido por regras, dividido em níveis de dificuldade e que acaba com um resultado quantificável [6].

Um jogo, para ser considerado como tal, deve seguir uma porção de requisitos que incluem: ser um sistema formal baseado em regras, possuir resultados variáveis e quantificáveis, os esforços emitidos pelos jogadores devem ser capazes de modificar o resultado, os jogadores devem se sentir ligados aos resultados, e as consequências da atividade podem ser opcionais [7].

Crawford [8] lista a existência de quatro elementos fundamentais aos jogos, sendo eles: Representação - o jogo fornece uma representação da realidade, mesmo que essencialmente subjetiva; Interação - o ponto crucial da representação previamente citada está na forma como ela interage com o jogador, que deve ser capaz de provocar alterações na realidade apresentada; Conflito - ao buscar atingir o objetivo do jogo, o jogador se deparará com obstáculos que poderão impedir que este objetivo seja alcançado facilmente; e Segurança - o ato de jogar permite que o jogador se submeta à experiência psicológica do conflito sem o risco de danos físicos.

Por um longo período os jogos foram tidos como atividades sem significado funcional, porém, a partir de estudos modernos, pode-se perceber que estes são importantes e até necessários em processos de aprendizagem, desenvolvimento e estimulação de

---

\*e-mail: ianmaiscedo@gmail.com

habilidades [9]. Os jogos estão diretamente ligados ao desenvolvimento humano. Através do jogo pode-se, mesmo sem perceber, adquirir novos conhecimentos e desenvolver grande parte das funções cognitivas [10].

## 2.2 Ensino de Lógica de Programação

A atividade de programação pode ser entendida como a construção de um mecanismo para resolver determinado problema [11]. Ainda segundo os autores, para programar, busca-se a elaboração de um conjunto de ferramentas computacionais para auxiliar em diversas fases, tais como: compreensão do problema; planejamento da solução e; construção do algoritmo.

Almeida et al. [11] citam a dificuldade no ensino de lógica de programação causada pelo desinteresse dos alunos nas disciplinas que abordam tal conteúdo acarretando em dificuldade no aprendizado. Os autores propõem o uso de ferramentas de uso simples e com características atrativas para os alunos como soluções para transpor o desafio de ensinar lógica de programação.

Com isso, deve-se diminuir a prática de fazer algoritmos no papel sem a oportunidade de efetivamente experimentá-lo em execução de determinada forma. Nascimento [3] cita materiais lúdicos, softwares dinâmicos e de fácil compreensão, além do uso de jogos como veículos que facilitam este processo de divulgação de conteúdo.

## 2.3 Compiladores

Compiladores podem ser entendidos como ferramentas que, a partir da leitura de um programa escrito em uma linguagem (linguagem fonte), realizam a tradução para uma outra linguagem (linguagem objeto) mantendo a semântica original [12]. Ao longo do processo de tradução, o compilador realiza duas tarefas básicas, são elas: análise, etapa em que o texto-fonte é analisado e compreendido em seus aspectos semânticos e síntese, na qual o texto-objeto é gerado para que possa corresponder ao texto de entrada.

Em termos de compiladores, o tradutor recebe um programa-fonte, realiza um processo de análise léxica (onde recebe um programa fonte e identifica os tokens de acordo com a gramática), após este, realiza um processo de análise sintática (onde recebe a saída da análise léxica e utiliza-a em conjunto com as regras de produção), e finalmente faz a síntese, onde será gerada a saída da tradução.

As Gramáticas Livres de Contexto (GLC) são as mais importantes na área de compiladores e linguagens de programação pelo fato de especificarem eficientemente as construções sintáticas usuais. Uma gramática pode ser definida por  $G = (N, T, P, S)$  onde  $N$  representa o conjunto de símbolos não-terminais,  $T$  representa o conjunto de símbolos terminais,  $P$  representa o conjunto de regras de produção e  $S$  representa o nó inicial. a GLC é uma gramática em que suas regras de produção seguem o modelo  $A \rightarrow \alpha$ , sendo  $A$  um não-terminal qualquer e  $\alpha$  um terminal ou um não-terminal qualquer [13].

A partir de tais considerações, este projeto foi desenvolvido com o objetivo de criar um protótipo do jogo “As Aventuras Espaciais de Cody”, um serious game onde o jogador, para realizar as devidas ações deverá digitar comandos básicos que serão submetidos a um processo de tradução e resultarão nos movimentos do personagem, ou seja, para jogar o jogador deverá - ainda que com comandos básicos - programar. Desta forma, o jogador aprenderá lógica de programação enquanto se diverte em um game.

## 3 MATERIAIS E MÉTODOS

Para o desenvolvimento deste jogo, foram utilizadas tecnologias como HTML5 para a construção da página e o elemento canvas, que utiliza um modelo procedural de bitmaps, podendo assim ser utilizado para desenhar imagens por meio de códigos. Paralelamente, foi utilizado o CSS3 para estilização da ferramenta. A linguagem de programação JavaScript foi responsável pela dinâmica e construção do jogo através da sua biblioteca PhaserJs que fornece recursos simples e funcionais que proporcionam a construção de jogos [14].

Para o processo de tradução dos códigos informados pelo jogador, será utilizado o JISON, um framework JavaScript que se encarrega deste processo de maneira fácil. O JISON recebe como entrada uma GLC e gera um arquivo capaz de fazer o processo de tradução da linguagem descrita pela gramática [15]. A GLC foi desenvolvida de modo a ser semelhante à sintaxe de linguagens de programação - incluindo conceitos de orientação a objetos - e será melhor descrita nas seções seguintes.

## 4 GAME DESIGN

O Game Design descreve a dinâmica e interação do jogo e foi dividido em duas seções que apresentam os elementos essenciais para a compreensão do jogo que será chamado de “As Aventuras Espaciais de Cody”, sendo eles: “O Jogo” e “Dinâmica de uma partida”.

### 4.1 O Jogo

A história se passa em um mundo imaginário onde depois do governo de “Bugu, o Terrível” o planeta Compilatatus vive em completa miséria financeira. A partir de tal dificuldade, o povo Compilatiano elege o jovem Cody para partir em uma odisséia espacial em busca das moedas intergaláticas - as mais valiosas de todo o universo - pois somente após reunir todas as moedas intergaláticas, Compilatatus poderá se salvar da crise financeira.

Ao todo, para juntar todas as 50 moedas intergaláticas existentes, o jogador terá que passar por três fases sendo que, a primeira delas será apenas de treino e nas demais encontrará inimigos e quantidades de moedas diferentes. Apesar de três fases terem sido planejadas para a versão final do jogo, por se tratar de um protótipo, este trabalho implementou somente a primeira fase. As fases do jogo estão descritas na Figura 1.

Fase / Planeta	Inimigos	Quantidade de Moedas
Training Room	Nenhum	50
Nulolândia	Fantasma Null	35
Tipolândia	Monstros Var	15

Figura 1: Relação entre fases, inimigos e quantidade de moedas.

As fases ocorrerão na sequência demonstrada na tabela, portanto, a primeira - que será executada ao iniciar o jogo - será a Training Room, fase onde o jogador irá simplesmente treinar e aprender a jogar. Na sequência, será jogada a Nulolândia, onde o jogador terá menos moedas e alguns Fantasmas estarão dispostos na tela, estes fantasmas ficarão parados, porém, ao tocá-los, o jogador perde o jogo, o que aumenta o desafio no jogo. Por último, Cody voará até Tipolândia onde encontrará apenas 15 moedas e os monstros Var, que irão rondar a área e, caso toquem Cody, o jogo acaba.

### 4.2 Dinâmica de uma partida

Ao iniciar o jogo, assim como demonstrado pela Figura 2, será apresentada uma tela onde o usuário poderá jogá-lo.



Figura 2: Tela de jogo Training Room.

Abaixo do canvas onde o jogo é executado haverá o campo de entrada de texto onde o jogador deverá digitar os comandos para que o personagem realize ações. O cenário apresentado na Figura 2 é o Training Room. Neste cenário não há inimigos e o objetivo é coletar cinquenta moedas, que vão aparecendo em séries de cinco moedas.

Após recolher todas as moedas de uma fase, Cody entra em sua nave e parte para o próximo planeta onde terá novos desafios e buscará mais moedas. Ao final do jogo será mostrada uma tela parabenizando o jogador / Cody por ajudar o povo Compilatiano a se livrar da miséria.

## 5 ESTRUTURA INTERNA

Para que os comandos digitados pelo jogador possam resultar em ações da aplicação, como a execução de uma determinada função, por exemplo, foi preciso construir uma gramática. A gramática criada para este projeto está representada na Figura 3

de acordo com a notação EBNF [16], onde palavras maiúsculas representam não-terminais e palavras em minúsculo representam terminais.

```

PROGRAMA ::= COMANDO ';'
COMANDO ::= 'cody' '.' FUNCAO
FUNCAO ::= 'andar' '(' DIRECAO ')' | 'pular' '(' DIRECAO? ')' | 'parar' '(' ')'
DIRECAO ::= 'direita' | 'esquerda'
    
```

Figura 3: Gramática dos comandos do jogo.

Na gramática mostrada na Figura 3 o não-terminal “PROGRAMA” representa o nó inicial da gramática. A partir deste não-terminal, qualquer coisa que o usuário digitar será considerado um comando. Partindo do nó não-terminal “COMANDO”, há três diferentes regras válidas, sendo elas “cody.andar(DIRECAO);”, “cody.pular();”, “cody.pular(DIRECAO);” e “cody.parar();”. Em ambas a palavra “cody” representa um nó terminal, assim como a palavra “parar”. As possibilidades partindo deste nó são representadas na Figura 4.

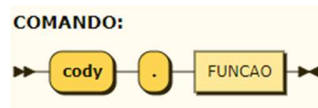


Figura 4: Diagrama representativo do não-terminal “COMANDO”.

Como demonstrado na Figura 4, todo comando deve iniciar com “cody” seguido por “.” e ter a seguir o não-terminal “FUNCAO”. O não-terminal, por sua vez pode resultar em qualquer um dos terminais “pular”, “andar” e “parar” que serão exibidos na Figura 5.

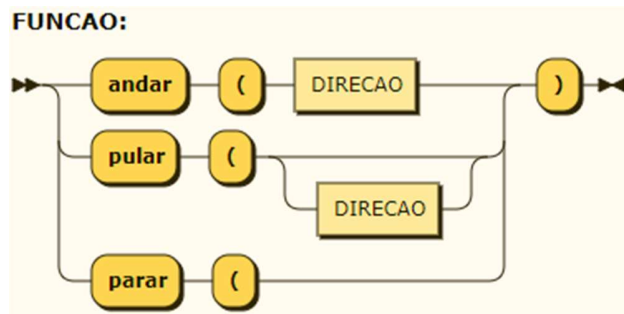


Figura 5: Diagrama representativo do não-terminal “FUNCAO”.

O não-terminal “DIRECAO” pode resultar em qualquer um dos dois terminais “direita” ou “esquerda”. Estes correspondem à direção a qual o Cody deverá se mover (andando ou pulando).

De tal forma, caso o jogador digite por exemplo: “cody.andar(direita);”, o personagem do jogo andará em direção à parte direita da tela até que o jogador dê o comando “cody.parar();” ou algum outro comando, desta forma também será trabalhada no jogador a ideia de laços de repetição.

## CONCLUSÃO

Este trabalho teve o objetivo de criar um protótipo do jogo “As Aventuras Espaciais de Cody”, um jogo sério onde o jogador, para realizar as devidas ações deverá digitar comandos básicos

que serão submetidos a um processo de tradução e resultarão nos movimentos do personagem, ou seja, para jogar o jogador deverá - ainda que com comandos básicos - programar. Desta forma, o jogador aprenderá lógica de programação enquanto se diverte em um game.

Para que o objetivo do jogo descrito neste trabalho fosse alcançado, diversos conceitos de diferentes áreas tiveram que ser interligados, como conceitos referentes a compiladores/tradutores, gramáticas, jogos e educação. Com isto, foi possível identificar possibilidades de transmitir o conteúdo de disciplinas de introdução a programação e orientação a objetos por meio de mídias com formato e estética diferentes das mais convencionais (jogos), que podem aumentar o nível de interesse dos acadêmicos e, conseqüentemente, o engajamento ao longo das disciplinas.

O protótipo criado e apresentado neste trabalho traz os principais pontos do jogo “As Aventuras Espaciais de Cody”, ainda sem integração com banco de dados e sem uma implementação completa. Portanto, como trabalhos futuros sugere-se a implementação do restante das fases do jogo, assim como desenvolvimento de um sistema de armazenamento de resultados e rankings baseados nos resultados. Além destes, sugere-se o planejamento de novas fases ou novas aventuras dentro deste mesmo universo que exercitem mais conceitos relacionados à programação, como possibilitar aos jogadores o controle de outros elementos do jogo, além do personagem Cody. Também como trabalho futuro, com o objetivo de avaliar o jogo proposto neste trabalho enquanto ferramenta de ensino, planeja-se utilizá-lo integrado ao plano de ensino de uma turma introdutória de programação no ensino superior, e avaliar o desempenho dos integrantes da turma na disciplina antes e depois da aplicação do jogo.

#### AGRADECIMENTOS

Os autores são gratos à Antônio Carlos P. M. Milhomens Júnior pelo auxílio durante a construção dos níveis planejados para implantação no jogo.

#### REFERÊNCIAS

- [1] OLIVEIRA, L. R. Metodologia do Desenvolvimento: um estudo de criação de um ambiente de e-learning para o ensino presencial universitário. Vale do Rio dos Sinos: Unisinos, 29 set. 2006.
- [2] MORATORI, Patrick Barbosa. Por que utilizar jogos educativos no processo de ensino aprendizagem? 2003. 33 f. Dissertação (Mestrado) - Curso de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003.
- [3] NASCIMENTO, Cledison da Silva. Introdução ao Ensino de Lógica de Programação para Crianças do Ensino Fundamental com a ferramenta Scratch. 2015.
- [4] WEILLER, Thaís Arrias. Game design inteligente: elementos de design de videogames, como funcionam e como utilizá-los dentro e fora de jogos. Escola de Comunicações e Artes da Universidade de São Paulo, [s.l.], p.1-157, 21 maio 2013. Universidade de São Paulo Sistema Integrado de Bibliotecas - SIBiUSP. <http://dx.doi.org/10.11606/d.27.2012.tde-17052013-105240>.
- [5] HUIZINGA, J. Homo ludens: o jogo como elemento da cultura. 5o. ed. [S.l.]: Perspectiva, 2003. p. 256.
- [6] LUCCHESI, Fabiano; RIBEIRO, Bruno. Conceituação de jogos digitais. Sao Paulo, 2009.
- [7] JULL, J., Half-Real: Video Games between Real Rules and Fictional Worlds, The MIT Press, 2005, ISBN: 0262101106.
- [8] CRAWFORD, C. (1982). The Art of Digital Game Design, Washington State University, Vancouver, 1982.
- [9] FORTALEZA, Sandra Maria; CONSOLARO, Marina Mancini. ESTIMULAÇÃO DAS MÚLTIPLAS INTELIGÊNCIAS POR MEIO DE JOGOS EDUCATIVOS EM CRIANÇAS DA 3a. SÉRIE. 2007.
- [10] ANACLETO, Junia C. et al. Ambiente para criação de jogos educacionais de adivinhação baseados em cartas contextualizadas. In: WORKSHOP SOBRE INFORMÁTICA NA ESCOLA, 28., 2008, Belém do Pará. Anais do XXVIII Congresso da SBC. Belém do Pará: Sbc, 2008. p. 29 - 38.
- [11] ALMEIDA, Eliana S. de et al. AMBAP: Um Ambiente de Apoio ao Aprendizado de Programação. In: CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 22., 2002, Florianópolis. Anais. Florianópolis: Sbc, 2002. p. 1 - 10. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2002/006.pdf>>. Acesso em: 19 jun. 2018.
- [12] SCHNEIDER, Carlos Sérgio; PASSERINO, Liliana Maria; OLIVEIRA, Ricardo Ferreira de. Compilador Educativo VERTO: ambiente para aprendizagem de compiladores. Renote: Novas Tecnologias na Educação, Porto Alegre, v. 3, n. 2, p.100-110, nov. 2005.
- [13] FURTADO, Olinto José Varela. LINGUAGENS FORMAIS E COMPILADORES.
- [14] FAAS, Travis. An Introduction to HTML5 Game Development with Phaser.JS. CRC Press, 2016.
- [15] CARTER, Zach. JISON: An API for creating parsers in JavaScript.
- [16] W3C. Extensible Markup Language (XML) 1.0 (Fifth Edition). 2008. Disponível em: <<https://www.w3.org/TR/xml/>>. Acesso em: 30 jul. 2018.