

Desenvolvendo o Pensamento Computacional através da Máquina de Turing: o enigma do Curupira

Fernanda Pires¹, Karolayne Batista Teixeira¹, Marcela Pessoa¹, Polianny Lima¹

¹Escola Superior de Tecnologia – Universidade do Estado do Amazonas (EST/UEA)
69.050-020 – Manaus – AM – Brazil

{kbt.lic, fpires, mspessoa, palima}@uea.edu.br

Abstract. *This paper presents the prototype of a serious game called "Curupira Machine" that aims to present the concept of the Turing Machine in a playful way. This is inspired in a scenario that emphasize the environmental responsibility. In the scenario, the main character has the mission to rescue caged animals and, for this, needs to decipher the codes using a Turing Machine. The game it was developed following a puzzle strategy, for the Android platform and third-person perspective. In solving the puzzle, the player develops Computational Thinking through the creation of strategies and algorithms to solve the problems proposed..*

Resumo. *Este artigo apresenta um protótipo de um jogo de propósito sério denominado "Máquina do Curupira" cujo objetivo é apresentar, de forma lúdica, o conceito de Máquina de Turing (MT), inspirado em um cenário que desperta a responsabilidade ambiental. No enredo, a personagem principal tem a missão de resgatar animais enjaulados e, para isso, precisa decifrar os códigos utilizando uma máquina de Turing. O jogo é desenvolvido seguindo uma estratégia do tipo Puzzle, para plataforma Android e com perspectiva em terceira pessoa. Ao solucionar o enigma, a jogador desenvolve o Pensamento Computacional através da criação de estratégias e algoritmos para a resolução dos problemas propostos.*

1. Introdução

Vivemos em tempos acelerados, em que as respostas aos problemas devem ser rápidas e eficientes. Cientistas e educadores defendem que o século XXI trouxe consigo maiores possibilidades de protagonizar o processo de resolução de problemas, e com isso, uma série de requisições no que tange habilidades e competências, entre elas conhecer sobre computação (McGonigal, 2017; ResnickandRobinson, 2017).

Dentre essas habilidades e competências, o termo “Pensamento Computacional” ganhou importância e visibilidade nos cenários de discussão desde o discurso de Wing (2006). Embora não exista um conceito formado sobre o tema, pesquisadores concordam que o conteúdo vem sendo discutido há muito tempo, tendo nomes como “Pensamento Algorítmico” e “Raciocínio Lógico” (Brackmann, 2017). Com as discussões sobre o inatismo do Pensamento Computacional, se buscam formas de desenvolver habilidades responsáveis pela resolução de problemas, como as utilizadas ao projetar algoritmos desde a infância, quando se consolidam importantes memórias de longo prazo (Pires et al., 2018; WingandStanzione, 2016).

A preocupação com a inserção da rubrica Pensamento Computacional na escola fez com que o currículo de países como EUA, Reino Unido e Austrália o incorporassem

à sua grade curricular (Leiteandda Silva, 2017), sendo muitos deles voltados especificamente para computação. Pensamento Computacional não diz respeito somente a Computação, mas a todas as áreas do conhecimento. Posto isto, este trabalho apresenta uma ferramenta para auxílio do desenvolvimento de Pensamento Computacional a partir daquela que marcou o começo de tudo, na computação, a máquina de Turing.

Ferramentas e métodos de aprendizagem devem ser desenhados respeitando as características cognitivas humanas, que podem sofrer variações de acordo com a faixa etária, características étnicas e sociais (ClarkandMayer, 2016). As tecnologias digitais vêm mudando a forma como as pessoas processam informação (Carr, 2011). As novas formas de aprender incluem o uso de ferramentas digitais que, por já estarem incorporadas à vida diária, diminuem a curva de aprendizagem. Nesse cenário, os jogos são apontados como ferramentas capazes de promover aprendizagem significativa, pela inerência divertida de suas estruturas e sua capacidade de promover engajamento (dos Santos *et al.*, 2018; Gee, 2003).

Este artigo apresenta o desenvolvimento de um jogo educacional denominado “Máquina do Curupira”, no estilo *Puzzle 2D*, inspirado na Máquina de Turing. Tem como base a teoria cognitiva da aprendizagem de Jean Piaget, o Construtivismo. Seu objetivo é auxiliar o desenvolvimento do Pensamento Computacional através da utilização da mecânica implícita, do funcionamento da MT. O jogo possui uma temática ambiental, possibilitando a conscientização sobre a caça ilegal e tráfico de animais em risco de extinção. A Máquina do Curupira tem a proposta de incentivar o indivíduo a criar estratégias lógicas para salvar animais através da codificação de uma MT. Os símbolos utilizados são frutas, pela familiaridade das crianças.

Desse modo, o artigo está organizado na seguinte estrutura: na Seção 2 estão os trabalhos relacionados; na Seção 3 está a fundamentação teórica utilizada no desenvolvimento do jogo; na Seção 4, a metodologia de desenvolvimento, resumo do *Game Design Document* – GDD do jogo e o conteúdo de aprendizagem proposto, na Seção 5 são apresentados os resultados esperados e na Seção 6 as considerações finais.

2. Trabalhos relacionados

Apesar da comunidade científica dedicar esforços ao desenvolvimento de Jogos Educacionais, ainda é escasso o número de propostas que utiliza como tema conteúdos relacionados à Teoria da Computação. Neste contexto, Nicolau e Pimentel (2018) apresentam um meio de utilizar jogos de tabuleiros em seus processos de criação, elaboração e jogabilidade, como forma de desenvolvimento do Pensamento Computacional. Eles registram que jogos de tabuleiro acompanham a evolução e foi utilizado, inclusive, por Charles Babbage, Alan Turing e Edward Thorp para explorar aspectos computacionais.

Já, Bombasar (2015) traz uma revisão sistemática que descreve ferramentas para aprendizagem do pensamento computacional de 2006 a início de 2015. Os autores questionam a falta de uma ferramenta que apresente o formalismo da teoria da Máquina de Turing, sugerindo que uma ferramenta inspirada neste conceito pode favorecer a criação de desafios para explorar a noção de computabilidade. Seguindo o sugerido por Bombasar (2015), este trabalho propõe uma jogo educacional cuja temática está relacionada a conteúdos de Teoria da Computação, especificamente, a utilização de Máquina de Turing como forma de apoiar o desenvolvimento do Pensamento

Computacional, explorando o cenário Amazônico e dando visibilidade à necessidade da preservação e manutenção de animais em extinção.

3. Pensamento Computacional, brincadeiras e desenvolvimento humano

O termo “Pensamento Computacional”, como todo tema em investigação, vem sofrendo modificações ao longo do tempo. Em 2006 Wing (2006) o definiu como “uma habilidade fundamental a ser usada por todos no século XXI”, por isso deveria ter um espaço nas escolas, assim como a escrita e a Matemática. A Computação, nesse contexto, seria a automação da abstração, assim também como defendido por Aho e Ulmam (1994). Pensamento Computacional, segundo Wing, envolve a solução de problemas, que pode ser através da projeção de sistemas ganhando um espaço especial a Ciência da Computação.

Computer Science Teachers Association e a *International Society for Technology in Education* (CSTA e ISTE) definiram o Pensamento Computacional como um processo de resolução de problemas em que suas características de formulação permitem que computadores, ferramentas ou pessoas possam compreendê-los e ajudar a resolvê-los. Incluiu como parte do processo a formulação de problemas computáveis, organização e análise da representação de dados, por meio de modelos e simulações, até a generalização de soluções otimizadas para um grande número de problemas (Barr et al., 2011).

Alguns estudos vêm sendo realizados em torno do inatismo do Pensamento Computacional como estruturas humanas inerentemente capazes de propor soluções a problemas ao longo do tempo. Partem do pressuposto de que tudo o que se tem no mundo tecnológico foi criado por pessoas, tais como: algoritmos, processos de otimização ou máquinas e é um grande desejo fazer com que máquinas simulem humanos (IA) (Soh et al., 2015; Wing, 2008; Worrell et al., 2015).

Mais recentemente foram definidos quatro pilares para trabalhar o desenvolvimento do Pensamento Computacional, sendo eles: i) decomposição, que diz respeito à forma como o problema pode ser dividido, ii) padrão, que trata da identificação de um padrão a partir da divisão anteriormente realizada, iii) abstração, que consiste em compreender os passos anteriores, descartando os elementos que não são necessários à proposição da solução e iv) algoritmo, que é a sequência que define a solução para o problema (BBC, 2018; Liukas, 2015).

3.1. Aprender brincando

Desde que se tem registros, as brincadeiras e os jogos tem um importante papel no desenvolvimento humano (dos Santos *et al.*, 2018; Huizinga, 1971; Rodrigues *et al.*, 2018; Soh et al., 2015), tendo um espaço específico nas Teorias de Aprendizagem cognitivas, que tem o homem como o centro do desenvolvimento a partir de suas construções. Os jogos e brincadeiras encontram suporte na teoria construtivista de Jean Piaget, por enxergar a aprendizagem como um processo que envolve organização, adaptação, assimilação e acomodação de conhecimentos (Piaget, 1976). Ao cruzar os pressupostos da Teoria Construtivista com os Pilares do Pensamento Computacional, é notório que a organização dos pensamentos é um processo importante para compreensão do problema, os passos definidos podem auxiliar na composição de métodos de aprendizagem eficientes, sobretudo através de jogos (Association, 2016; dos Santos et al., 2018; Huizinga, 1971; MichaelandChen, 2005).

A teoria construtivista de Piaget versa sobre a possibilidade que tem o indivíduo de construir seus próprios conhecimentos, participando ativamente do processo e enfatizando a importância do erro (Fino, 2004). Uma das características dos jogos é a possibilidade de se manter engajado em solucionar os problemas conforme são apresentados, o erro ganha uma conotação motivadora e tentar não é sinal de fracasso e sim de aumento da expertise (Huizinga, 1971; McGonigal, 2017; MichaelandChen, 2005). Os jogos se mostram como alternativas lúdicas à aprendizagem. As brincadeiras fazem parte do processo de construção do sistema cognitivo humano, pelos sistemas lógicos e simbólicos relacionados na formação de memórias referentes à resolução de problemas (De Freitas, 2018).

3.2. Máquina de Turing aplicada à Máquina do Curupira

Esta seção apresenta os conceitos de Máquina de Turing, descrevendo o jogo proposto e a relação entre os elementos do jogo e a definição de Sipser (2007):

3.2.1 Máquina de Turing - MT

Máquina de Turing é um modelo abstrato, proposto pelo matemático inglês Alan Turing em 1936, tão poderoso que até hoje não se conhece outra máquina com maior poder computacional (Vieira, 2006). Em sua essência, toda MT move-se ou move símbolos, de uma posição para outra em uma fita infinita. Ela tem um cabeçote de fita que pode ler e escrever símbolos e mover-se sobre a fita. A fita contém apenas a cadeia de entrada e está em branco no restante. O cabeçote de leitura-escrita pode mover-se tanto para esquerda quanto para a direita. A máquina continua a computar até que ela decida produzir uma saída. As saídas são obtidas entrando em estados designados de aceitação e rejeição, caso contrário continua infinitamente, nunca parando (Hopcroft *et al.*, 2003).

Segundo Sipser (2007), uma máquina de Turing é uma 7-upla, $(Q, \Sigma, \Gamma, \delta, q_0, q_{aceita}, q_{rejeita})$, onde Q, Σ, Γ são conjuntos finitos e:

1. Q é o conjunto de estados,
2. Σ é o alfabeto de entrada sem o símbolo em branco \sqcup ,
3. Γ é o alfabeto de fita, onde $\sqcup \in \Gamma$ e $\Sigma \subseteq \Gamma$,
4. $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D\}$ é a função de transição,
5. $q_0 \in Q$ é o estado inicial,
6. $q_{aceita} \in Q$ é o estado de aceitação, e
7. $q_{rejeita} \in Q$ é o estado de rejeição, onde $q_{rejeita} \neq q_{aceita}$.

A seção seguinte explica o funcionamento do jogo proposto fazendo uma relação entre a definição formal descrita nesta seção e os elementos utilizados no jogo.



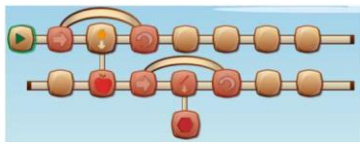

3.2.2 A Máquina do Curupira

Máquina do Curupira é um jogo digital educacional de estratégia do tipo *Puzzle*, para plataformas móveis, com perspectiva em terceira pessoa e câmera fixa. A personagem principal é baseada em uma lenda folclórica amazônica, o Curupira. O jogo se propõe, também, a auxiliar na conscientização sobre o tráfico e a caça ilegal de animais, especialmente os em risco de extinção. A dinâmica do jogo acontece em um cenário cujo objetivo é libertar os animais de jaulas. Cada jaula tem um código a ser decifrado, em uma analogia à Máquina de Turing. O jogo, ajuda a desenvolver responsabilidade

ambiental, divulga uma lenda amazônica e apoia o desenvolvimento do Pensamento Computacional para solucionar problemas por meio da exploração do conceito da MT.

O jogo possui os princípios básicos de uma MT, que são, o cabeçote, o movimento para esquerda e direita, leitura e escrita. As frutas correspondem ao alfabeto e o estado inicial depende da fase em que o jogador se encontra. O estado de aceitação ou rejeição está condicionado à validação da palavra de entrada (sequência de frutas), sendo o estado final da Máquina do Curupira i) aceitação: quando a palavra lida for igual a de entrada e ii) rejeição caso contrário. A Tabela 1 apresenta os elementos da definição formal da Máquina de Turing, relacionados aos elementos utilizados no jogo.

Tabela 1. Relação entre a definição formal de MT e os elementos do jogo.

7-upla da Máquina de Turing	Elementos correspondentes no jogo
1. Q é o conjunto de estados	Está implícito é visualizado durante a execução do jogo.
2. Σ é o alfabeto de entrada sem o símbolo em branco \sqcup	
3. Γ é o alfabeto de fita, onde $\sqcup \in \Gamma$ e $\Sigma \subseteq \Gamma$	
4. $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D\}$ é a função de transição	
5. $q_0 \in Q$ é o estado inicial	Depende do estado de cada fase.
6. $q_{aceita} \in Q$ é o estado de aceitação	Uma sequência aleatória de frutas 
7. $q_{rejeita} \in Q$ é o estado de rejeição, onde $q_{rejeita} \neq q_{aceita}$.	Quando o usuário gera uma sequência diferente da sequência de fruta de aceitas.

O jogo segue um modelo de aprendizagem progressiva, iniciando com tarefas semipreenchidas e a cada fase o nível de dificuldade aumenta, apoiando o desenvolvimento dos quatro pilares do Pensamento Computacional através da i) decomposição do código para encontrar o melhor meio de decifrar, ii) do reconhecimento dos padrões buscando minimizar o número de passos na execução da solução, iii) a abstração da solução rápida para que o jogador não perca suas conquistas e iv) elaboração de um algoritmo para solucionar o problema e libertar os animais.

4.1. Processo de Desenvolvimento

O processo de Desenvolvimento da “Máquina do Curupira” ocorreu em etapas que são: i) idealização, ii) especificação de requisitos do jogo, iii) planejamento, iv) prototipação e v) testes, conforme descritas a seguir:

Levantamento de requisitos: através de *brainstorm* definiu-se questões sobre Pensamento Computacional e a Máquina de Turing para realização do projeto.

Especificação de Requisitos do Jogo: foram especificados os objetivos da ação e como a fundamentação pedagógica e teórica podem sustentar a proposta de solução pensada, bem como a documentação do processo. Foram definidos como requisitos: delimitação do público alvo, objetivos de aprendizagem e limitações pedagógicas.

Planejamento: foi desenvolvido o *Design* de aprendizagem atrelado ao *Game Design Document – GDD*, nele foi definido a história do jogo, a personagem principal, *gameplay*, o universo do jogo, as mecânicas do *gameplay*, os inimigos, os materiais bônus, controles, interface, a progressão de *gameplay*, o gráfico de ritmo e o cronograma.

Prototipação: o primeiro protótipo em papel (baixa fidelidade) deve ser usado para realizar os testes de usabilidade, jogabilidade, bem como, as questões de interação e reconhecimento da interface. Se escolhe a *game engine* para desenvolver a próxima etapa e o protótipo de alta fidelidade é criado. A linguagem escolhida foi C#, a *game engine* Unity e o jogo está em desenvolvimento. A Figura 1 ilustra o processo de prototipação.

Testes (produto final): são executados testes para validar as etapas anteriores, caso não haja contrapontos, o produto pode ser entregue para utilização.

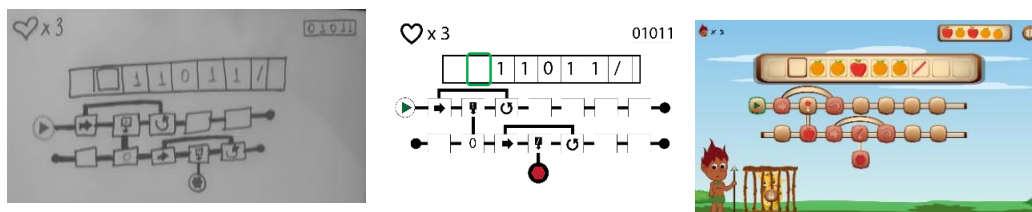


Figura 1. Protótipo no Papel, Adobe Illustrator e no PowerPoint.

4.2. A proposta do jogos e seus elementos de aprendizagem

Cenário: o jogo A Máquina do Curupira possui uma visão 2D, câmera em terceira pessoa com ângulo plano. Desenvolvida na *Engine* de jogos Unity, na Linguagem de Programação C# e sistema operacional Windows 10. Na versão atual, o jogo possui 3 níveis. A cada nível o jogador desbloqueia uma nova habilidade. A proposta do jogo é que o usuário aprenda a utilizar as funções sem necessitar de tutorial. A cada nível ele vai desbloqueando novas funções da máquina, auxiliando a abstração, solucionando os problemas lógicos e ajudando a desenvolver o Pensamento Computacional.

Storytelling: o protetor da Floresta Amazônica, o Curupira, tem a missão de proteger a fauna e a flora contra caçadores e homens maus, porém existem muitos perigos para manter a floresta segura. O Curupira, ao passear pela floresta encontrou animais que foram capturados e presos em jaulas. Todos são animais ameaçados de extinção como, onça pintada, macaco prego, peixe-boi, entre outros. As jaulas utilizam trancas protegidas por um código. Para libertar os animais, é necessário seguir um algoritmo baseado na MT e resolver o enigma. A cada jaula aberta, uma espécie de animal é salva dos caçadores. O jogo é ambientado na Floresta Amazônica, os símbolos e signos selecionados devem proporcionar imersão ao usuário. A mecânica do jogo foi construída tendo como base o funcionamento da MT, a jogabilidade permite que o usuário desenvolva o raciocínio lógico através da resolução dos problemas apresentados.

Mecânica: a “Máquina do Curupira” exige um certo grau de abstração. Respondendo a isso, os níveis iniciais dos algoritmos apresentados estão parcialmente incorretos, para que o usuário possa realizar a correção. A cada nível concluído novas funcionalidades são desbloqueadas permitindo a auto regulação da aprendizagem. Ao

iniciar o primeiro nível, o jogador é direcionado para tela de *storytelling*, apresentadas na Figura 2. Após essas telas o jogador conhece o animal que terá que resgatar, através de histórias rápidas. A jornada do herói é executada pelo jogador, que fica imerso na missão definida pelo *storytelling*. O *feedback* é implícito, como a posição do cabeçote, informando em qual passo do algoritmo ele está. É permitido ao usuário visualizar o *status* do sistema, situando o jogador sobre a quantidade de “vidas” restantes. O jogador possui ainda, acesso ajuda na tela de *pause* sobre o nível em que se encontra.



Figura 2. Apresentação do *Storytelling*

Os *power-up* coletados, representados por vidas e estrelas, proporcionam vantagem dentro do jogo, geralmente coletados após uma conquista do jogador, como resolver o problema em menor tempo ou menor quantidade de tentativas e testes. A Tabela 2, apresenta a organização dos personagens, elementos que configuram a mecânica e suas funções no cenário do jogo.

O controle do jogo ocorre através de quatro tipos de botões, conforme Tabela 3. O jogador deve combiná-los para encontrar um algoritmo de solução. No jogo os desafios encontrados pelo jogador devem ser resolvidos em poucos passos e a cada tentativa frustrada de destrancar a jaula ele perde uma vida. A cada nível o jogador encontra um problema mais difícil. Para passar um nível é necessário ter ao menos uma das três estrelas recebidas no início do jogo, considerando uma média entre o tempo gasto e a solução apresentada.

Tabela 2. Elementos do Jogo











Elementos	Nome	Função
	Curupira	Personagem Principal
	Animais	A serem resgatados
	Frutas	Alfabeto da Máquina
	Estrelas	Desbloquear próximo nível
	Vida Extra	Mais 1 tentativa

Tabela 3. Controle do jogo

Teclas	Funcionalidade
	Escrita (Troca a Fruta)
	Leitura (Condicional)
	Direção (Esquerda ou direita)
	Iniciar e Parar o Algoritmo
	Repetição

4.3. Algoritmo da Solução das Fases

A proposta de máquina de Turing ludificada segue os quatro pilares do Pensamento Computacional (decomposição, padrão, abstração e algoritmo). Considerando que o usuário precisa primeiro decompor o problema a ser solucionado ao verificar cada um dos símbolos separadamente. Observa-se um padrão de resolução à medida que os passos são executados. Abstrai-se uma possível solução a partir dos passos executados anteriormente e finalmente o usuário consegue organizar mentalmente um algoritmo capaz de chegar à solução do problema apresentado.

<p>Nível1()</p> <pre> cadeia Cabecote = "Vazio" inteiro Posicabecote = -1 cadeia lista[6] = {"Laranja","Laranja","Maca","Laranja","Laranja", "FIM"} cadeia listaCerta[6] = {"Maca","Laranja","Maca","Laranja","Laranja","FIM"} Posicabecote++ // movercabecoteDireita Cabecote = lista [Posicabecote] se (Cabecote == "Laranja") lista[Posicabecote] = "Maca" enquanto (Cabecote != "FIM") Posicabecote++ // movercabecoteDireita Cabecote = lista[Posicabecote] se (Cabecote == "FIM") para (inteiro j = 0; j<=5; j++) escreva(lista[j], "\n") escreva("\n") para (inteiro j = 0; j<=5; j++) escreva(listaCerta[j], "\n") para (inteiro i = 0; i <=5; i++) se (lista[i] == listaCerta[i]) escreva ("igual \n") se (i==6) escreva ("Fim da lista; Resposta certa \n") senão escreva("errado \n") </pre>	<p>Nível2()</p> <pre> cadeia Cabecote = "Vazio" inteiro Posicabecote = -1 cadeia lista[6] = {"Laranja","Laranja","Maca","Laranja","Maca", "FIM"} cadeia listaCerta[6] = {"Maca","Laranja","Maca","Laranja","Laranja","FIM"} Posicabecote++ // movercabecoteDireita Cabecote = lista [Posicabecote] se (Cabecote == "Laranja") lista[Posicabecote] = "Maca" enquanto (Cabecote != "FIM") Posicabecote++ // movercabecoteDireita Cabecote = lista[Posicabecote] se (Cabecote == "FIM") Posicabecote-- // movercabecoteEsquerda Cabecote = lista[Posicabecote] lista[Posicabecote] = "Laranja" Posicabecote++ // movercabecoteDireita Cabecote = lista[Posicabecote] para (inteiro j = 0; j<=5; j++) escreva(lista[j], "\n") para (inteiro j = 0; j<=5; j++) escreva(listaCerta[j], "\n") para (inteiro i = 0; i <=5; i++) se (lista[i] == listaCerta[i]){ escreva ("igual \n") se (i==6) {escreva ("Fim da lista; Resposta certa \n"); }senao { escreva("errado \n")} </pre>

Figura 4. Algoritmo de solução do nível 1 e 2.

A estrutura de solução do problema possibilita ao jogador exercitar estruturas condicionais, repetições, variáveis, organização lógica da proposição de listas bem como a organização de algoritmos. A Figura 4 representa as duas primeiras fases e seus respectivos algoritmos. O algoritmo do Nível 1 faz com que alguns estados fiquem bloqueados induzindo o jogador à solução do problema. A estrutura de dados que representa a fita nas duas fases é uma lista com as opções percorrer, ler e escrever. O Nível 2 não apresenta estados bloqueados fazendo com que o jogador desenvolva sua própria estratégia de jogo (algoritmo).

5. Conclusão

Muito se tem discutido sobre ferramentas digitais que possam auxiliar no desenvolvimento de estruturas cognitivas responsáveis pela otimização de processos de resolução de problemas. O artigo apresentou uma proposta de jogo educacional que busca auxiliar no desenvolvimento do Pensamento Computacional através de uma Máquina de Turing intitulada “A Máquina do Curupira”. O jogo está em processo de desenvolvimento, tendo sido concluído a mecânica, o projeto de *design* gráfico e de

aprendizagem e o *Game Design Document*. Atualmente está em fase de implementação. Na etapa de prototipação de baixo nível foram feitos testes de validação de usabilidade e jogabilidade. Diante dos resultados, foram realizados ajustes na interface, como por exemplo, o tamanho do botão de execução, a orientação da tela, o formato da apresentação da fita, o sistema de *feedback* na execução do algoritmo e a localização das personagens.

Como trabalhos futuros serão implementados novos níveis, realizadas avaliações de IHC, incluindo testes de Percurso Cognitivo, Usabilidade de Nielsen, Requisitos Educacionais, Heurística, TCAM, teste *GameFlow* e um relatório específico de percurso cognitivo. Todos esses testes serão aplicados com o público alvo, professores e *designers*.

6. Referências

- Aho, A. V., & Ullman, J. D. (1994). *Foundations of Computer Science: C Edition*: W. H. Freeman.
- Association, C. S. T. (2016). Operational definition of computational thinking.
- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23.
- BBC. (2018). KS3 Bitesize Computer Science: Introduction to computational thinking. Retrieved 17/05/2019, 2018, from <https://www.bbc.com/bitesize/guides/zp92mp3/revision/1>
- Bombasar, J., Raabe, A., de Miranda, E. M., & Santiago, R. (2015). *Ferramentas para o ensino-aprendizagem do pensamento computacional: onde está Alan Turing?* Paper presented at the Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE).
- Brackmann, C. P. (2017). Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica.
- Carr, N. (2011). *The shallows: What the Internet is doing to our brains*: WW Norton & Company.
- Clark, R. C., & Mayer, R. E. (2016). *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*: John Wiley & Sons.
- De Freitas, S. (2018). Are games effective learning tools? A review of educational games. *Journal of Educational Technology & Society*, 21(2), 74-84.
- dos Santos, A. L., Maurício, R. d. A., Dayrell, M., & Figueiredo, E. (2018). *Exploring Game Elements in Learning Programming: An Empirical Evaluation*. Paper presented at the 2018 IEEE Frontiers in Education Conference (FIE).
- Fino, C. N. (2004). Convergência entre a teoria de Vygotsky e o construtivismo/construcionismo. *acesado em*, 23(11), 2016.
- Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1), 20-20.
- Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2003). *Introduction to Automata Theory, Languages, and Computation*: Pearson Education International.

- Huizinga, J. (1971). *Homo ludens: o jogo como elemento da cultura* (Vol. 4): Editora da Universidade de S. Paulo, Editora Perspectiva.
- Leite, M., & da Silva, S. F. (2017). *Redimensionamento da Computação em Processo de Ensino na Educação Básica: O pensamento Computacional, o Universo e a Cultura Digital*. Paper presented at the Anais dos Workshops do Congresso Brasileiro de Informática na Educação.
- Liukas, L. (2015). *Hello Ruby: adventures in coding* (Vol. 1): Macmillan.
- McGonigal, J. (2017). *A realidade em jogo*: Editora Best Seller.
- Michael, D. R., & Chen, S. L. (2005). *Serious games: Games that educate, train, and inform*: Muska & Lipman/Premier-Trade.
- Nicolau, M., & Pimentel, L. (2018). Os jogos de tabuleiro e a construção do pensamento computacional em sala de aula. *Temática, 14*(11).
- Piaget, J. (1976). The grasp of consciousness: Action and concept in the young child.(Trans by S. Wedgwood).
- Pires, F. G., Duarte, J. C., da Silva Pessoa, L., dos Santos Pereira, K. S., Melo, R., & de Freitas, R. (2018). *Uma análise cognitiva entre a emergência de padrões em narrativas infantis e elementos do pensamento computacional*. Paper presented at the Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE).
- Resnick, M., & Robinson, K. (2017). *Lifelong Kindergarten: Cultivating Creativity Through Projects, Passion, Peers, and Play*: MIT Press.
- Rodrigues, P., Souza, M., & Figueiredo, E. (2018). Games and Gamification in Software Engineering Education: A Survey with Educators.
- Sipser, M. (2007). *Introdução à teoria da computação*: THOMSON PIONEIRA.
- Soh, L.-K., Shell, D. F., Ingraham, E., Ramsay, S., & Moore, B. (2015). Learning through computational creativity. *Commun. ACM, 58*(8), 33-35.
- Vieira, N. J. (2006). *Introdução aos fundamentos da computação: linguagens e máquinas*: Pioneira Thomson Learning.
- Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725.
- Wing, J. M., & Stanzione, D. (2016). Progress in computational thinking, and expanding the HPC community. *Communications of the ACM, 59*(7), 10-11.
- Worrell, B., Brand, C., & Repenning, A. (2015). *Collaboration and Computational Thinking: A classroom structure*. Paper presented at the 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).