

Desenvolvimento de um Jogo de Tabuleiro para Estimular Habilidades Relacionadas ao Pensamento Computacional

Alisson Steffens Henrique
Universidade do Vale de Itajaí
Itajaí, SC
ali.steffens@gmail.com

Rodrigo Lyra
Universidade Federal de Santa Catarina
Florianópolis, SC
rodrily@gmail.com

Andre Luis Maciel Santana
Universidade de São Paulo
São Paulo, SP
andrelms91@gmail.com

ABSTRACT

The ability to computationally think is considered one of the most important skills for citizenship in the 21st century. Considering that students are motivated when using games to learn and that formal education tends not to be fun and often less challenging. This paper implemented a board game capable of stimulating skills learning related. The developed game is a story-based board game RPG in which in-battle actions are programmed by the player. It was possible to notice the increase in players' abstraction and debugging capacity, especially when played cooperatively.

KEYWORDS

Pensamento Computacional, Jogos, Educação, Tabuleiro, Habilidades, Avaliação

1 INTRODUÇÃO

Em uma sociedade caracterizada por incertezas e mudanças rápidas, a capacidade de pensar de maneira criativa vem se tornando a chave para o sucesso e satisfação pessoal e profissional [1].

Além disso, o anseio por uma economia baseada em conhecimento e a grande procura por profissionais nos ramos de tecnologia e desenvolvimento, repercutem em um aumento na busca de talentos na área da Computação [2]. De acordo com Yardi e Bruckman [3], o problema nesta busca está no fato de que muitos estudantes pensam, equivocadamente, que a Ciência da Computação se restringe ao conhecimento referente à navegação na internet, processamento de textos ou realização de cálculos matemáticos.

Por outro lado, os que procuram estudar Ciência da Computação, logo descobrem que aprender a programar não é uma tarefa fácil, dada a necessidade da compreensão de conceitos abstratos. Além disso, grupos de estudantes costumam ser grandes e heterogêneos, o que dificulta aos professores o entendimento das dificuldades de cada estudante [4]. Somando isso ao fato de que em muitas vezes os estudantes não sabem o que esperar do curso, é comum que ao chegarem na matéria introdutória de algoritmos acabem reprovando ou abandonando a disciplina [5].

O conceito de Pensamento Computacional é, de acordo com Brennan e Resnick [6], uma maneira de se expressar que permite novos métodos de resolver problemas, e que mesmo se associando em vários momentos à programação, não se restringe à mesma.

Segundo Wing [7], o Pensamento Computacional é uma maneira de resolver os problemas e a concepção de sistemas e compreensão do

comportamento humano que se baseia em conceitos fundamentais da Ciência da Computação. No intuito de amenizar esta dificuldade inicial de aprendizado apresentada por Lahtinen [4], é comum o uso de atividades de ensino que envolvam entretenimento para o desenvolvimento de Pensamento Computacional nos alunos.

Sabendo da importância da Ciência da Computação na economia atual, e do Pensamento Computacional na compreensão e resolução de problemas, Blikstein [8] afirma que o Pensamento Computacional é uma das habilidades mais importantes para o exercício da cidadania no século XXI.

Para Bell [9], uma solução para o problema de falta de conhecimento em computação é a utilização de jogos educativos. Mais especificamente projetos como o CSUnplugged.org, um site com várias atividades introdutórias a conceitos de computação.

Estas atividades expõe a Ciência da Computação a crianças, sem a necessidade da utilização de computadores além de não serem atividades que conduzam as pessoas a simplesmente simular execuções de processadores, visto que de acordo com o autor, este tipo de atividade poderia se tornar tediosa.

Quanto a isto, Koster [10] mostra que o engajamento do jogador é baseado na capacidade do jogo em lhe propor desafios que não sejam nem muito fáceis, nem impossíveis, mas que o façam buscar por padrões para resolução das tarefas do jogo, diferentes dos padrões com os quais ele já está acostumado.

Conforme Papastergiou [11], estudantes sentem-se mais motivados se utilizando jogos para aprender, o mesmo ainda afirma que o ensino formal tende a ser tedioso e pouco desafiador, mas que com a utilização de jogos é possível um feedback imediato quanto aos erros e acertos, possibilitando aos estudantes validar hipóteses e aprender com suas ações.

Clua e Bittencourt [12] adentram ainda mais esta ideia e dividem os jogos educativos em duas partes, os de entretenimento e os didáticos. E afirmam que normalmente, jogos que são desde sua concepção planejados com a intenção de educar, costumam não ser tão divertidos. Por outro lado, alguns jogos como Age of Empires, SimCity 2000 e Grim Fandango, mesmo que feitos para serem apenas divertidos, acabam trazendo alguns conceitos de gestão de recursos, geografia, história e cultura que poderiam ser utilizados no escopo escolar.

Além do fato de jogos de entretenimento podem ser educativos, Kamii e DeVries [13] ainda demonstram que atividades em grupo tendem a ser mais eficientes no aprendizado, deste modo, é possível

afirmar que jogos educacionais com dinâmicas de multijogador tendem a ser mais eficientes do que os para um jogador.

Estas ideias foram amadurecidas por Papert [14] [15], que prega a construção do conhecimento, de modo colaborativo e baseado em projetos. Mais tarde, Resnick [16], inseriu o termo Aprendizagem Criativa, demonstrando a importância dos quatro P's, que conforme a Figura 2 tratam de projetos, parcerias, paixão e pensar brincando para a aprendizagem. Os quatro P's prezam para que os projetos desenvolvidos no ensino sejam significativos para os estudantes, de maneira divertida e em colaboração com colegas.

De acordo com CGI [17], muitas escolas brasileiras ainda não contam com a infraestrutura necessária (seja computadores bons para jogos ou até mesmo a falta de internet) para que jogos digitais possam ser utilizados.

Por isso, a proposta de Bell [2] é investir nas já citadas atividades desplugadas, em que além de descartar a necessidade de Internet, computadores e até mesmo energia elétrica para suas aplicações, podem ser ainda mais eficientes que as digitais por mostrarem ao aluno que os problemas resolvidos pela Ciência da Computação vão muito além de programar. Além disso, Bell ainda defende que atividades desplugadas tendem a ser mais facilmente aplicadas, sendo que não é necessário um ambiente específico com ferramentas complexas como computadores ou instalação prévia dos recursos necessários para sua aplicação.

Tüzün [18], demonstra que o uso de jogos no ensino pode melhorar o desempenho do estudante, enquanto Santana [19], afirmam que aliando o entretenimento de um jogo com táticas de ensino construcionistas, pode-se criar uma ferramenta divertida e que ensine.

Raabe [20], mostra algumas ferramentas que são normalmente utilizadas para o ensino de programação à iniciantes, e conclui que dentre estas, a mais divertida e que possibilita maior engajamento e motivação nos alunos é o jogo Light-bot.

Em Light-bot o jogador deve solucionar problemas em que o robô deve acender luzes em um ambiente isométrico com projeção ortogonal, por meio de lógica algorítmica. À medida em que os problemas vão sendo solucionados, o jogador é encaminhado para novos desafios onde esta lógica pode ser reforçada ou novos conceitos podem ser apresentados.

Gomes [21] aponta que o jogo Light-Bot apresenta uma série de problemas relacionados à faixa etária de sua pesquisa, no entanto, ao relacionar o projeto com a pesquisa proposta neste documento, é possível considerar que o principal problema é o fato de o jogo ignorar o conceito de loops, apresentando apenas a recursividade como solução para este tipo de problema. Além disso, para aplicação de atividades com o Light-bot é necessário um dispositivo com acesso à internet para cada estudante.

Outras atividades e ferramentas desplugadas também foram verificadas, como as de Bell [2], o scratch cards, potato pirates, entre outros, que permitirão a construção da Seção sobre computação desplugada. Tanto no mercado, quanto na literatura existem poucas ferramentas que possibilitam a introdução à computação de maneira divertida, desplugada, com notações simples e que seja português.

Este projeto propôs um jogo de tabuleiro baseado em RPGs (Jogos de Interpretação de papel) de mesa, educacional, não didático, para 2 a 4 jogadores, que tem como público-alvo estudantes na fase final do ensino médio ou com o mesmo concluído. O jogo deve permitir que os professores o imprimam, sendo assim, mesmo em escolas com um menor orçamento para este tipo de material a atividade poderá ser realizada.

2 Metodologia

Buscando avaliar o jogo proposto, foi utilizada uma metodologia baseada na de Moreno-León [22], que encontrou convergência entre relações sociais e a melhoria na capacidade de programar dos estudantes. O trabalho em questão analisa códigos feitos no Scratch, e para calcular uma pontuação relacionada à complexidade do algoritmo criado pelo estudante, utiliza de uma ferramenta chamada Dr. Scratch.

Dr. Scratch é uma ferramenta de avaliação de complexidade de algoritmo desenvolvida com base no trabalho de Boe [23], onde, a partir do código fonte de um programa em Scratch, a ferramenta é capaz de pontuar o algoritmo com base nos blocos utilizados no projeto e considerando interatividade, sincronismo, lógica, representação de dados, controle de fluxo e outras instruções avançadas. Considerando a diferença de comandos possíveis entre os programas em Scratch e os do jogo desenvolvido, esta tabela será modificada conforme demonstrada no Quadro 1.

Categoria	Comandos valendo		
	1 ponto	2 pontos	3 pontos
Movimentação	avançar, virar à esquerda / direita	-	-
Batalha	Atacar, defender	Usar Item, Usar Magia	-
Repetição	-	repita 2 vezes, repita 3 vezes	para cada aliado/inimigo em sua visão
Desvio Condicional	-	se (não?) houver algum inimigo adjacente	se o personagem estiver com mais/menos de 3 pontos de vida

Quadro 1: Pontuação por comando

Moreno-León [22], também destaca como importante fator no desenvolvimento dos estudantes, o quanto a comunicação ou sociabilidade pode permitir que eles melhor desenvolvam seus conhecimentos em programação.

Lopes [24], utiliza uma avaliação qualitativa baseada em observação de partidas no seu jogo de tabuleiro para poder determinar a motivação e diversão dos estudantes durante o aprendizado. Com estas observações, ele consegue identificar e quantificar, o quanto a interação dos estudantes influenciou no resultado da partida.

Sobre a metodologia de Lopes [24], são avaliadas as ações de interação entre jogadores, entre o jogador e o jogo, do jogador com ele mesmo, e ações fora do jogo. Tendo isso, foi proposta no Quadro 2, uma sequência de possíveis ações a serem observadas durante uma partida do jogo desenvolvido nesta pesquisa, na intenção de poder de

alguma forma mensurar também a relação entre a socialização dos estudantes com o resultado das batalhas.

Interações	
Planejando	
Observar Personagem	Jogador observa a posição do personagem, sua ficha e suas características
Observar Mapa	Jogador observa o mapa, a localização dos outros jogadores, dos inimigos e dos obstáculos
Observar Inimigos	Jogador observa a posição do Inimigo, sua ficha e suas características
Planejar	Jogador pensa no jogo sem necessariamente interagir com o mesmo
Observar Regras	Jogador lê o livro de regras
Executando	
Depurar Personagem	Jogador tenta prever os próximos movimentos do seu personagem, e valida seu código
Depurar Inimigo	Jogador tenta prever as próximas ações do inimigo
Executar	Jogador executa o código de seu personagem durante a fase de Execução
Interagindo com o Grupo	
Discutir com o Mestre	Jogador comenta ou pergunta ao mestre sobre as regras, inimigos ou a história do jogo
Cooperar	Jogador discute com os outros jogadores, os próximos passos a serem tomados por um indivíduo ou pelo grupo, bem como possíveis estratégias de jogo
Divagando	
Repousar	Jogador parado, sem mexer ou olhar para o jogo, e não demonstrando interesse no mesmo
Discussão não cooperativa	Jogadores conversam sobre coisas não relacionadas ao jogo
Viajar	Jogador utiliza celular para fins não relacionados ao jogo, ou demonstra não interesse no mesmo, porém fazendo alguma coisa alternativa
Interagindo com o Personagem	
Equipar Itens	Jogador equipa itens no personagem, sem ser nos espaços de programação
Programando	
Codificar	Jogador coloca cartas de ação ou itens na área de código da ficha de personagem

Quadro 2: Ações Possíveis

A intenção da análise destes comportamentos foi gerar indícios para o estabelecimento de perfis dos estudantes, para que assim fosse possível obter dados sobre a evolução dos jogadores durante a partida, bem como validar o desenvolvimento ou utilização das habilidades citadas como estruturais no Pensamento Computacional.

Quanto às habilidades de Pensamento Computacionais descritas na Fundamentação Teórica e que podem ser desenvolvidas ou utilizadas em cada estado, destacam-se as a seguir:

Planejando ou Interagindo com o Grupo: formulação de problemas de forma a ser possível utilizar um computador ou outra ferramenta para ajudar a resolvê-los, a organização e análise lógica de dados,

representação de dados através de abstrações, identificação e análise de possíveis soluções com o objetivo de alcançar a combinação mais eficiente e eficaz, e a generalização de alguns processos de resolução de problemas para uma variedade mais ampla.

Executando: automatização de Soluções.

Interagindo com o Personagem ou Programando: organização e análise lógica de dados, representação de dados através de abstrações, implementação de soluções em busca de combinações mais eficientes e eficazes e automatização de soluções por meio de pensamento algorítmico.

Divagando: não desenvolve habilidades relacionadas ao Pensamento Computacional.

Além disso, foi feita para cada turno uma tabela para calcular a complexidade dos algoritmos definidos pelos jogadores em cada turno.

3 O Jogo

Antes de iniciar o desenvolvimento do jogo (ferramenta de aprendizagem), foram definidos os conteúdos relacionados ao pensamento computacional a serem abordados no jogo, para que, conhecendo-os pudesse ser elaborado o conjunto de instruções possíveis nas ações do jogador.

Para isso foi feita uma relação entre os conceitos abordados pelos jogos voltados ao desenvolvimento de habilidades relacionadas ao Pensamento Computacional, bem como as utilizadas no MOD de Minecraft desenvolvido por Santana [19], em busca dos conceitos de programação mais utilizados.

Tendo em vista que os conceitos que mais aparecem são: Logo (* utilização da linguagem para movimentação de personagem em grade de mapa), laços de repetição, desvio condicional e variáveis, foi optado por utilizar destes conceitos no jogo. Estes conceitos (condições e repetições) são também citados por Raabe [25] como os assuntos a serem abordados na disciplina de Pensamento Computacional no Ensino Médio, e por Boe [23] para o cálculo de complexidade dos algoritmos com a ferramenta Dr. Scratch.

Foi definido então que o personagem deveria movimentar-se pelo mapa utilizando instruções semelhantes às de movimentação da linguagem LOGO, podendo colocar estas instruções dentro de desvios condicionais ou laços de repetição. As variáveis seriam utilizadas nas condições de desvio e repetição podendo ser referente ao personagem, aliados, inimigos ou ao ambiente. Sobre a utilização de funções, acredita-se que seja possível, porém para jogadores mais experientes e acostumados com as mecânicas definidas neste projeto, por isso foi optado por não ser uma mecânica nesta versão inicial.

As mecânicas do jogo foram definidas com base nos Trabalhos Similares. Grande parte da jogabilidade foi influenciada diretamente pelos RPGs de mesa como Dungeons & Dragons®. Este projeto se focou no desenvolvimento de uma mecânica de batalhas que possa ser utilizada não somente com o RPG nele desenvolvido, mas também com outros já existentes.

Esta mecânica de batalhas consiste na utilização de cartas como forma de programar as ações do personagem durante a batalha.

A primeira coisa a ser feita é escolher uma missão a ser cumprida pelos jogadores, o mestre deve então posicionar os objetos e inimigos

do mapa de acordo com o descrito na missão. Em seguida, cada jogador deve selecionar o personagem que deseja utilizar para jogar, considerando para isso suas especialidades e características. Na etapa seguinte, os jogadores devem preparar suas fichas de personagem, posicionando as fichas de contador de vida nos locais indicados pelas mesmas, bem como colocando suas miniaturas no local indicado no mapa.

A Ficha de personagem bem como os demais *assets* do jogo são descritos a seguir:

3.1 Ficha de Personagem

Baseada no tabuleiro de codificação de Code Master, a ficha de personagem contém não somente informações sobre características do personagem, que serão utilizadas para tomada de decisão e definição de valores durante a parte de interpretação do jogo, mas também uma área de codificação chamada de Ações do Turno.

Como pode ser visto na Figura 1, as cartas de ação devem ser posicionadas na seção Ações do Turno, para que sejam executadas durante a fase de execução do personagem. Além disso, algumas variáveis também estão presentes como, Pontos de Vida (PV), e os atributos (vitalidade, sabedoria, força, inteligência, sorte e agilidade) que tem valores diferentes dependendo do personagem.



Figura 1: Ficha de Personagem

O inventário foi baseado no do jogo, onde o personagem pode usar duas armas de uma mão ou uma de duas mãos.

3.2 Cartas de Movimentação

Inspiradas na linguagem logo, as cartas de movimentação seguem padrões parecidos com os demonstrados em Bits & Bytes e Robot Turtle, permitindo que o jogador movimente o personagem em uma grade (o mapa), utilizando os comandos Avançar, Virar à Direita e Virar à esquerda, conforme visto na Figura 2.



Figura 2: Cartas de Movimentação

3.3 Cartas de Repetição

As cartas de repetição (Figura 3) são baseadas nas “cartas de guia” de Code Monkey Island e permitem ao jogador repetir uma determinada ação do personagem uma quantidade específica de vezes, ou de acordo com a situação dos outros personagens.



Figura 3: Cartas de Repetição

3.4 Cartas de Desvio Condicional

Baseadas nos jogos Code Monkey e Code Master as cartas de desvio condicional (Figura 4) permitem ao jogador impor condições para que o personagem realize ou não alguma ação, baseado em características dos personagens ou do ambiente.



Figura 4: Cartas de Desvio Condicional

3.5 Cartas de Batalha

As cartas de batalha (Figura 5) servem para permitir que o personagem ataque um adversário, ou se prepare para se defender de um possível

ataque do oponente. Existem algumas cartas de itens que também podem ser utilizadas no lugar das cartas de Atacar ou Defender, e as cartas de magia ainda podem ser utilizadas em conjunto com as de ataque para um turno mais complexo.



Figura 5: Cartas de Batalha

3.6 Cartas de Itens

Assim como em Munchkin e a grande maioria dos RPGs, o personagem pode portar itens que modificam suas características, ou o permitiram atacar e defender de maneira diferente do padrão. As Cartas de Itens (Figura 6), podem ser referentes a armas de uma mão, escudos, ou armas de duas mãos, e algumas delas podem ainda ser utilizadas como Cartas de Batalha, se tal informação estiver em seu texto.



Figura 6: Cartas de Itens

3.7 Cartas de Personagem

Além de se movimentar, atacar e defender, o personagem pode também fazer ações ainda mais impactantes no jogo, as magias (Figura 7), que podem curar, mudar temporariamente atributos de personagens, ou até mesmo alterar as ações programadas de algum personagem. Após utilizadas, estas cartas têm um tempo de resfriamento, e só podem ser utilizadas após ficarem este número de turnos sem o ser.



Figura 7: Cartas de Magia

A cada turno, o personagem com menor iniciativa é o primeiro a programar (normalmente um inimigo). Ele deve então posicionar as cartas em seus devidos slots e passar o turno para o próximo jogador até que todos tenham posicionado suas cartas em ordem. Estas ações podem servir para movimento, ataque, defesa, cura, interação com objetos ou interação com outros personagens, utilizando as cartas já apresentadas. Em sequência, segue-se a ordem de iniciativa do menor para o maior.

Após todos terem definidos suas ações, são executadas as ações do último jogador a programar (o de maior iniciativa), seguidas pelas do anterior, até que o último personagem (normalmente algum inimigo) realize suas ações.

As mecânicas do jogo foram definidas com base nos Trabalhos Similares. Grande parte da jogabilidade foi influenciada diretamente pelos RPGs de mesa como Dungeons & Dragons, e este trabalho se focou no desenvolvimento de uma mecânica de batalhas que possa ser utilizada não somente com o RPG nele desenvolvido, mas também com outros já existentes.

O jogo utiliza dados com diferentes quantidades de faces. O mais comum durante a aventura é o d20. Em casos onde o resultado de alguma ação é incerto, os jogos de RPG costumam recorrer aos d20 para determinar sucessos ou fracassos. Todos os personagens contêm pontos de atributos, que são: vitalidade, sabedoria, força, inteligência, sorte e agilidade. Estes atributos influenciam nas rolagens de dados como modificadores de resultados.

Quando um valor 20 é tirado em um d20, é considerado um crítico e normalmente nestas situações, o personagem não somente consegue executar a ação como também ganha alguma vantagem para o jogo, que pode variar de acertar algum outro inimigo no mesmo ataque até receber informações extras ou itens que auxiliem no decorrer da aventura.

Por outro lado, quando um número muito baixo é rolado em um d20, o personagem pode receber uma desvantagem, como se machucar, perder sua arma, acertar um amigo, ou estragar alguma negociação.

Os personagens só podem atacar alvos em seu campo de visão. Para verificar o campo de visão, é necessário analisar duas condições: alcance e linha de visão. O alcance é definido considerando os pontos de alcance do personagem. Para um personagem com 3 pontos de alcance por exemplo, estão em seu alcance todos os tiles que podem

ser alcançados com até 3 movimentos ortogonais à partir de sua posição atual.

A linha de visão de um personagem consiste em: para cada tile no alcance do personagem criar uma linha imaginária conectando o centro do tile onde está o personagem e o centro do tile em questão. Caso esta linha passe por algum outro objeto ou uma parede, este tile não está no campo de visão do personagem. Um exemplo pode ser observado na Figura 8.



Figura 8: Campo de Visão

A Figura 8 demonstra as linhas de visão do personagem. Neste caso, traçando as linhas imaginárias entre os tiles, percebe-se que a árvore acaba ocultando a visão de quatro tiles que estão no alcance do personagem, mas não em seu campo de visão (os tiles destacados em laranja). Isto deve-se ao fato de ao traçar a linha de visão, a linha passar pelo tile da árvore, o que de acordo com as regras do jogo, faz com que o personagem não o veja mais.

4 Testes

Após definidas todas as mecânicas bem como a interface do jogo, foram feitos dois testes com a versão física impressa, utilizando três jogadores voluntários cada.

Em ambos os testes foi utilizada uma história pré-definida durante o desenvolvimento da Metodologia, os personagens foram apenas escolhidos pelos jogadores e não criados. Os testes duraram cerca de cinco horas cada e durante cada aventura houve duas batalhas. Foi feita uma análise com relação à complexidade dos algoritmos e as interações dos usuários.

Sobre a metodologia de Lopes [24], foram avaliadas as ações de interação entre jogadores, entre o jogador e o jogo, o jogador com ele mesmo, e ações fora do jogo, considerando as subdivisões, conforme exibido no Quadro 3.

Interações	
Interagindo entre jogadores	
Jogar	Perguntar, responder, comentar sobre o jogo ou acumular atributos e pontuações
Socializar	Conversar, rir
Falar sobre Regras	Conversar sobre as regras do jogo
Interação do Jogador com o jogo	
Preparar	Ordenar cartas, contar dinheiros, etc.
Mover	Mover uma peça, controlar um personagem, usar uma arma
Observar Regras	Estudar ou observar as regras
Observar Jogo	Observar o jogo enquanto os outros jogam
Interação do Jogador com ele mesmo	
Recordar	Recordar informação da memória
Decidir	Formular escolhas sobre movimentação
Interação fora do jogo	
-	Jogador não está prestando atenção no jogo

Quadro 3: Iterações Possíveis para Lopes

Pensando nisso, foi proposta no Quadro 2, uma sequência de possíveis ações a serem observadas durante uma partida do jogo desenvolvido nesta pesquisa, a fim de mensurar a relação entre a socialização dos estudantes com o resultado das batalhas.

Baseado neste modelo de coleta, foram feitos os testes, onde para cada turno de cada batalha, em cada aplicação, foi gerada uma imagem. a Figura 9 é uma destas imagens, e demonstra a posição inicial (no turno) de cada personagem, bem como o movimento a ser realizado naquele turno após a execução dos passos. Além disso, era destacado também a área de ataque do personagem ao final do turno.

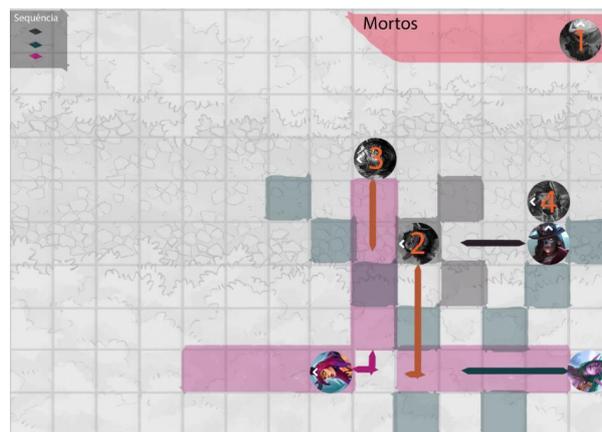


Figura 9: Descrição visual de um turno

A primeira aplicação foi experimentada com jogadores dos cursos de Ciência da Computação e Enfermagem. Em relação às interações dos jogadores durante a primeira batalha, foi possível notar um aumento no tempo de planejamento e de interação com o grupo, que repercutiu em uma melhora na qualidade dos algoritmos gerados. Após cerca de uma hora de interpretação de personagens, os jogadores entraram em uma segunda batalha.

Em uma avaliação voltada às interações dos jogadores, na segunda batalha, destaca-se o aumento no tempo de interação com o grupo e uma diminuição no tempo de Programação, sem muita perda em relação a qualidade dos algoritmos, o que pode ser considerada uma maior intimidade com a utilização deste recurso.

A segunda aplicação contou com a repetição de uma das jogadoras, e teve apenas jogadoras graduandas em enfermagem. Uma situação interessante observada no terceiro turno desta primeira batalha, foi o fato de em algum momento entre o segundo e terceiro turno, alguém ter esbarrado na mesa, o que gerou um deslocamento da uma personagem em diagonal. Como este deslocamento não foi percebido durante a batalha, acabou interferindo na movimentação futura da personagem, mas não influenciou de maneira negativa a experiência dos jogadores ou a história em nenhum aspecto.

Ainda sobre a avaliação de interações das jogadoras, foi possível notar que a cooperação e discussão relacionada a possíveis estratégias e movimentações teve um grande incremento durante esta batalha, o que refletiu na qualidade das jogadas, sendo que, fora o primeiro turno, praticamente não houve erros de lógica de programação durante este jogo.

Diretamente após a primeira batalha, houve ainda uma pausa no jogo para interação e alimentação, seguida da continuação da história. Esta pausa na aplicação mostrou-se positiva, e as jogadoras puderam descansar e recuperar energias para continuar a aplicação. Um ponto interessante é que os assuntos durante este intervalo foram quase que inteiramente voltados a estratégias para o jogo, o que indica certo grau de engajamento.

Em relação às interações das jogadoras, na segunda batalha, destaca-se um aumento ainda maior na fase de interação com o grupo e planejamento simultâneos, tomando decisões mais impactantes no resultado do jogo e praticamente sem divagar ou tratar de assuntos que não fossem relacionados ao jogo durante o mesmo

Foi analisada também, a complexidade dos algoritmos gerados pelos jogadores durante as rodadas. O quadro 4 exhibe estas complexidades em relação a um dos testes.

Complexidade	Batalha 1			Batalha 2	
	1	2	3	1	2
Lady M.	6	8	6	4	6
Jhannia	4	6	4	2	6
Rachel	4	7	7	4	5

Quadro 4: Complexidades Teste 2

Neste quadro é possível visualizar as diferenças entre a complexidade dos algoritmos (calculada conforme Quadro 1 desenvolvidos durante as duas batalhas realizadas no teste em questão. A complexidade, entretanto, não demonstrou relação com a qualidade do código, principalmente durante o jogo. Deste modo foi optado por não a considerar como fator determinante de aprendizagem.

5 Conclusões

O Pensamento Computacional, bem como as habilidades relacionadas ao mesmo, ainda passa por muitas dificuldades no que se diz respeito à definição, e em muitas vezes acaba sendo motivo de discussão entre

diversos autores sobre como melhor classificá-lo. É quase que unanimidade entretanto a afirmação de o desenvolvimento dele em estudantes de diversas faixas etárias é importante não somente para o indivíduo mas também para que se possa ter uma sociedade melhor, com uma maior qualidade de vida e bom uso das ferramentas tecnológicas trazidas pela computação para o mundo atual.

Pensando nisso, muitos países têm trazido este tema para o ensino básico, transformando seus métodos e grades de ensino em algo mais semelhante ao que é vivido pelos estudantes no mundo fora dos muros da escola. Em contrapartida, este movimento necessita que sejam criadas formas de ensino, como ferramentas e jogos, que possam fazer este tipo de conteúdo ser mais facilmente digerido por alunos de várias realidades diferentes.

O desenvolvimento deste tipo de ferramenta se mostrou necessário, e as pesquisas elaboradas durante o desenvolvimento deste projeto indicaram não somente que habilidades relacionadas ao Pensamento Computacional podem sim ser desenvolvidas de maneira eficiente com a utilização de jogos de tabuleiro como também que estes jogos podem ser divertidos e ainda assim cumprirem bem seu papel como ferramentas de aprendizado.

Os diversos tipos de teste permitiram ainda ter uma visão mais detalhada do comportamento dos estudantes durante o uso da ferramenta e mostraram ainda que para este tipo de validação, a análise da pura complexidade relativa dos algoritmos desenvolvidos pelos estudantes não representa de forma válida o aprendizado dos mesmos e que, por outro lado, o estudo e observação de suas interações durante a programação e execução podem trazer dados mais valiosos em relação a como cada algoritmo pode resolver ou não um problema.

A análise situacional de cada turno do jogo, considerando aspectos de otimização de código e planejamento tático de estratégias, trouxeram como principal colaborador da aprendizagem dos estudantes os seus próprios colegas, parceiros no jogo. Sendo assim, conclui-se que um jogo de tabuleiro cooperativo pode sim desenvolver habilidades relacionadas ao Pensamento Computacional, principalmente quando o feito de maneira colaborativa e construcionista ou construtivista.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

REFERENCIAS

- [1] R. Florida, *The Rise of the Creative Class--Revisited: 10th Anniversary Edition--Revised and Expanded*, 2 edition. Basic Books, 2012.
- [2] T. Bell, J. Alexander, I. Freeman, and M. Grimley, "Computer Science Unplugged: School Students Doing Real Computing Without Computers," *Journal of Applied Computing and Information Technology*, vol. 13, no. 1, pp. 20–29, 2009.
- [3] S. Yardi and A. Bruckman, "What Is Computing? Bridging the Gap Between Teenagers' Perceptions and Graduate Students' Experiences Categories and Subject Descriptors," *Proceedings of the third international workshop on Computing education research*, pp. 39–49, 2007.
- [4] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *ACM SIGCSE Bulletin*, vol. 37, no. 3, p. 14, 2005.
- [5] A. Petersen, M. Craig, and D. Zingaro, "Reviewing CS1 exam question content," *Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11*, p. 631, 2011.
- [6] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," *Annual American Educational*

- Research Association meeting, Vancouver, BC, Canada*, pp. 1–25, 2012.
- [7] J. M. Wing, “Computational thinking,” *Commun. ACM*, vol. 49, no. 3, pp. 33–35, 2006.
- [8] P. Blikstein, “O pensamento computacional e a reinvenção do computador na educação.” 2008.
- [9] T. Bell, J. Alexander, I. Freeman, and M. Grimley, “Computer science without computers: new outreach methods from old tricks,” *21st Annual Conference of the National Advisory Committee on Computing Qualifications*, pp. 127–133, 2008.
- [10] R. Koster, *A Theory Of Fun For Game Design*. 2013.
- [11] M. Papastergiou, “Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation,” *Computers and Education*, vol. 52, no. 1, pp. 1–12, 2009.
- [12] E. W. G. Clua and J. R. Bittencourt, “Uma Nova Concepção para a Criação de Jogos Educativos,” *Simpósio Brasileiro de Informática na Educação*, p. 36, 2003.
- [13] C. Kamii and R. DeVries, *Group Games in Early Education: Implications of Piaget’s Theory*. National Association for the Education of Young Children, 1834 Connecticut Avenue, N.W., Washington, DC 20009 (NAEYC Publication #317), 1980.
- [14] S. Papert, *Mindstorms*. 1980.
- [15] S. Papert, “Situating Constructionism,” *Constructionism*, pp. 1–11, 1991.
- [16] M. Resnick, “Give P’s a chance: Projects, Peers, Passion, Play,” *Constructionism and Creativity Conference*, 2014.
- [17] CGL.br/NIC.br, *TIC Educação*. 2016.
- [18] H. Tüzün, M. Yilmaz-Soylu, T. Karakuş, Y. Inal, and G. Kizilkaya, “The effects of computer games on primary school students’ achievement and motivation in geography learning,” *Computers and Education*, vol. 52, no. 1, pp. 68–77, 2009.
- [19] A. L. M. Santana and P. Martins, “Desenvolvimento e avaliação de modificação do jogo minecraft para estimular o pensamento computacional em estudantes do ensino médio,” *Workshops do Congresso Brasileiro de Informática na Educação*, no. Cbie, p. 92, 2017.
- [20] A. Raabe, G. Zanchett, and A. Vahldick, “Jogos de Programar como uma Abordagem para os Primeiros Contatos dos Estudantes com a Programação,” *Workshops do Congresso Brasileiro de Informática na Educação*, no. Cbie, p. 1485, 2015.
- [21] T. C. S. Gomes and J. C. B. de Melo, “O Pensamento Computacional no Ensino Médio : Uma Abordagem Blended Learning,” *XXI Workshop sobre Educação em Computação - WEI. Anais do XXXIII Congresso da Sociedade Brasileira de Computação - CSBC.*, no. July 2013, pp. 651–660, 2013.
- [22] J. Moreno-León, G. Robles, and M. Román-González, “Examining the Relationship between Socialization and Improved Software Development Skills in the Scratch Code Learning Environment,” *J. Univers. Comput. Sci.*, 2016.
- [23] B. Boe, C. Hill, M. Len, G. Dreschler, P. Conrad, and D. Franklin, “Hairball: Lint-inspired Static Analysis of Scratch Projects,” *ACM SIGCSE Bulletin*, 2013.
- [24] R. P. Lopes, “Cabinet-strategy board game for network and system management learning,” *XIII Simpósio Brasileiro de Jogos e Entretenimento Digital*, 2014.
- [25] A. Raabe, N. Ellery, E. A. de Jesus, E. Silva, J. Bombasar, and A. L. M. Santana, “A Experiência de Implantação de uma Disciplina Obrigatória de Pensamento Computacional em um Colégio de Educação Básica,” *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, vol. 6, no. 1, p. 1182, Oct. 2017.