

Desenvolvimento de um Jogo Digital Educacional para o Ensino de Pensamento Computacional Concorrente

Pedro Marques

*Dep. de Ciência da Computação, IM
UFRJ*

Rio de Janeiro, Brasil
pedromn@cos.ufrj.br

Eduardo Mangeli

*Eng. de Sist. e Computação, COPPE
Dep. de Ciência da Computação, IM*

UFRJ
Rio de Janeiro, Brasil
mangeli@cos.ufrj.br

Rafael Studart Monclar

*Eng. de Sist. e Computação, COPPE
UFRJ*

Rio de Janeiro, Brasil
rstudart@cos.ufrj.br

Geraldo Xexéo

*Eng. de Sist. e Computação, COPPE
Dep. de Ciência da Computação, IM*

UFRJ
Rio de Janeiro, Brasil
xexeo@cos.ufrj.br

Resumo—Este artigo apresenta o processo de concepção e desenvolvimento do Mapa do Tesouro, um jogo eletrônico para o ensino de Pensamento Computacional Concorrente, um conceito derivado do Pensamento Paralelo, que por sua vez, seria um componente do Pensamento Computacional apresentado por Wing. O jogo desenvolvido tem como público alvo crianças de idade entre 4 e 10 anos. Os conceitos do PC Concorrente foram incorporados em processos do jogo, com a utilização da teoria da Retórica Procedural. O jogo foi avaliado por trinta e nove crianças durante sua utilização em aulas de um curso extracurricular de Pensamento Computacional.

Palavras-Chave—game design, retórica procedural, pensamento computacional, jogos educacionais

I. INTRODUÇÃO

Este trabalho apresenta os processos de idealização, desenvolvimento e avaliação do jogo educacional Mapa do Tesouro, desenvolvido pelos autores para auxiliar crianças de 4 a 10 anos no aprendizado de Pensamento Computacional Concorrente (PC Concorrente), uma expansão proposta do conceito Pensamento Paralelo [1].

Pensamento Computacional (PC) é uma habilidade que permite utilizar conceitos de ciência da computação para solucionar problemas e analisar sistemas. Wing argumenta que essa é uma competência fundamental para todos, não só para cientistas da computação [2]. O desenvolvimento dessa competência foi incluído em currículos escolares de diversos países, com destaque para o Reino Unido [3].

Jogos são ferramentas populares para ensino de diversos temas [4]. No caso de Algoritmos, um dos principais tópicos do PC [1], é possível encontrar diversas aplicações lúdicas com este objetivo [5]. Entretanto, como será descrito nas próximas seções, alguns tópicos do PC ainda possuem carência de jogos educacionais.

Utilizamos a teoria da retórica procedural [6] durante a fase de idealização, de modo a articular a mensagem que se pretende transmitir e as mecânicas presentes no jogo.

Os conceitos de retórica procedural [6] e as definições dos componentes de jogos do framework MDA [7] foram utilizados na etapa de idealização do jogo e sustentaram a articulação entre sua mensagem e propósito *vis-à-vis* as mecânicas apresentadas.

Durante a etapa de desenvolvimento, utilizou-se um processo iterativo, com produtos intermediários e avaliações que permitiram correções e modificações no jogo para solucionar problemas e o aprimora. Este processo segue relatado neste trabalho, bem como o detalhamento dos elementos do jogo tais quais seus comandos, desafios e fases.

Ao final do processo, o jogo foi utilizado por 39 alunos, com idades entre 4 e 9 anos, que expuseram suas impressões a respeito da dificuldade encontrada no jogar, da disposição de repetir a experiência de jogar e de sua impressão sobre o prazer experimentado. As observações dos pesquisadores sobre as particularidades de cada grupo e uma análise qualitativa das respostas dos alunos está estruturada ao final deste trabalho.

II. REFERENCIAL TEÓRICO

Para contemplar os objetivos descritos anteriormente, o referencial teórico será dividido em três segmentos:

- O que é Pensamento Computacional e qual sua relevância pedagógica;
- O que é a Retórica Procedural e como ela sustenta a criação de processos educacionalmente interessantes em jogos; e
- A relação e a efetividade do uso de jogos como ferramenta educacional.

A. Pensamento Computacional

Wing apresentou o conceito Pensamento Computacional (PC), em 2006, descrevendo-o como a habilidade que permitiria analisar sistemas, solucionar problemas e entender o comportamento humano utilizando conceitos comuns da Ciência da Computação [2]. Contudo, ainda que esse artigo apresente diversas aplicações do PC, não há descrição completa de quais seriam seus principais tópicos, ou como estes deveriam ser ensinados. O interesse da comunidade acadêmica pela proposta fez surgir diversos estudos com novas ideias e possíveis respostas às principais questões levantadas.

Em 2013, sete anos depois, Grover e Pea revisitaram a literatura sobre PC e listaram os principais conceitos e definições que compõe a habilidade [1]. Uma das explicações destacadas é de autoria da *Royal Society*: “Pensamento Computacional é o processo de reconhecer aspectos computacionais do mundo em nossa volta e aplicar técnicas da Ciência da Computação para entender e analisar tanto sistemas artificiais quanto naturais”, em tradução livre. Neste mesmo artigo, Grover e Pea enumeram nove competências do PC:

- abstração e detecção de padrões;
- processamento Sistemático de Informação;
- sistemas simbólicos e representações;
- noções algorítmicas de fluxo de controle;
- decomposição de problemas;
- pensamento recursivo, iterativo e paralelo;
- lógica condicional;
- eficiência e desempenho; e
- depuração e detecção de erro.

Em plataformas de ensino, como BBC Bitesize [8], Code.Org [9] e Google for Education [10], outra abordagem se popularizou. Ela estrutura o PC a quatro pilares: abstração, decomposição, algoritmos e reconhecimento de padrões. Embora exista compatibilidade com o trabalho de Grover e Pea em alguns tópicos, conceitos como detecção de erro e pensamento paralelo têm menor destaque.

Kirkpatrick [11] argumenta sobre a importância do chamado Pensamento Computacional Paralelo relacionando-o com a habilidade de criar algoritmos que permitam a execução de várias instruções em simultâneo, de maneira menos centralizada que soluções tradicionais e mais adequada à realidade dos computadores atuais. Kirkpatrick cita professores da área de computação que destacam os benefícios de desenvolver essa capacidade nos alunos, que passam a criar soluções com múltiplos processos independentes, utilizando melhor os recursos do computador, em vez de um único processo que executa as instruções passo-a-passo e de maneira linear.

B. Retórica Procedural

A Retórica Procedural [6] é a prática de usar processos de maneira persuasiva, para mudar a opinião de uma pessoa ou modificar suas ações. Bogost considera que jogos possuem a capacidade de descrever (e modelar) o mundo através de

processos e, por conta disso, a Retórica Procedural seria sua maneira prática e efetiva de expressão [12].

Um exemplo utilizado por Bogost é o gênero de jogos de simulação de naves: seus processos apresentam ao jogador uma aproximação de como é a experiência de pilotar uma aeronave. Essa aproximação da realidade consegue produzir no jogador reflexões sobre a dificuldade (ou facilidade) de pilotar. Tornar o jogo mais realista, adicionando mais regras, influencia seus processos e pode incentivar o jogador a ser mais cauteloso [12].

C. Jogos e Aprendizado

Jogos são amplamente aceitos como uma importante ferramenta à disposição do educador para criar situações de aprendizagem [13]. De forma mais audaciosa, Prensky argumenta que jogos eletrônicos seriam o motor de uma revolução em curso no ensino, a geração de alunos e aprendizes que sempre experienciaram o mundo sob a onipresença da tecnologia acabaria por exigir a mudança das ferramentas utilizadas na educação e a efetiva utilização de jogos eletrônicos como recurso pedagógico [14].

Há uma clara ligação entre o que se deseja de uma situação de aprendizado e o que acontece com o jogador nos jogos, especialmente nos vídeo games. Entre as facetas positivas podemos citar, por exemplo, a motivação intrínseca e o engajamento gerados, e a imersão por longo tempo em um assunto, característica necessária para o aprendizado profundo [15]. Os jogos eletrônicos seriam a mídia ideal para promover o engajamento de estudantes nativos digitais e atender seu anseio de protagonismo no processo de aprendizagem [14]. Van Eck [4] afirma que não apenas esses estudantes podem valer-se de jogos eletrônicos como ferramenta de aprendizado, e que a efetividade do processo de aprendizado baseado em jogos eletrônicos, *Digital Game-Based Learning – DGBL*, pode ser empregada no aprendizado de temas diversos para alunos de diferentes níveis de inclusão tecnológica.

Em *A Theory of Fun*, Koster argumenta que jogos, mesmo de entretenimento, “servem como uma ferramenta fundamental e poderosa de aprendizado” [16], sendo essa, na verdade, uma das formas mais importantes de encontrar diversão em jogos.

Já há algum tempo, esse quadro tem feito com que fossem criados jogos eletrônicos educacionais. Possivelmente o jogo eletrônico educacional de maior impacto na indústria, e um dos maiores sucessos comerciais entre jogos eletrônicos de modo geral, foi *The Oregon Trail - Brøderbund* de 1985, baseado em um jogo textual com o mesmo nome de 1971, que buscava ensinar a crianças em idade escolar sobre a vida de pioneiros que se dirigiam ao Oregon no ano de 1847 [17].

Mais recentemente, com a larga distribuição de computadores pessoais, notebooks, *tablets* e *smartphones*, e com a geração nativa digital se impondo como um novo desafio educacional, o interesse por (vídeo) jogos educacionais aumentou muito [4]. Entretanto, os desafios de desenvolvê-los repetidamente e com sucesso ainda não foram resolvidos. Este artigo descreve um jogo desenvolvido dentro de um contexto

de estudo sobre como desenvolver sistematicamente jogos com propósito.

III. PENSAMENTO COMPUTACIONAL CONCORRENTE

Partindo do conceito de Pensamento Paralelo, listado por Grover e Pea [1], este trabalho propõe uma expansão ao qual atribuímos o nome de Pensamento Computacional Concorrente (PC Concorrente), intimamente ligado à ideia de Concorrência na Ciência da Computação.

Maziero descreve que na computação o problema da concorrência ocorre quando dois ou mais processos, executados de forma sobreposta no tempo, devem cooperar na manipulação de recursos compartilhados [18].

Pacheco [19] e Breshears [20] observam que os casos de concorrência se dividem em dois: de maneira intercalada, onde um processo começa e pode interromper sua execução para ceder a vez para o próximo; ou de uma maneira simultânea, na qual as tarefas executam suas instruções ao mesmo tempo em núcleos ou processadores diferentes. Os autores descrevem que paralelismo ocorre quando as tarefas tem suas instruções executadas simultaneamente. [19] Concorrência, então, seria um termo mais abrangente que pode se materializar como paralelismo [20].

A próxima seção define Pensamento Computacional Concorrente (PC Concorrente) tomando como base os conceitos expostos.

A. Definição

O PC Concorrente é a habilidade de planejar soluções e entender problemas que apresentam diversos agentes, cada um com seus algoritmos próprios que são executados de maneira sobreposta no tempo e podem ter dependências no uso de recursos compartilhados entre si.

Ensinar PC Concorrente é expandir o ensino de algoritmos, um dos pilares do PC, para diversos objetos com comportamentos distintos e que podem interagir entre si. Por conta dessa relação, o natural seria primeiro desenvolver uma base lógica nos alunos com algoritmos simples e depois introduzir outros personagens (ou processos).

IV. TRABALHOS CORRELATOS

Tendo em vista a definição proposta de Pensamento Computacional Concorrente, fizemos buscas com objetivo de encontrar *software* (jogos e aplicativos) que estivessem em sintonia pedagógica. As pesquisas foram feitas usando Google Scholar como mecanismo de busca e outros sites ligados a distribuição e venda de jogos.

A. Jogos

Monclar *et al.* [5] revisaram diversos jogos com propósito que tem como objetivo ensinar ou auxiliar o aprendizado de programação [5]. Desta lista, destacamos alguns jogos: Lightbot Hour [21], Cato's Hike [22], Move the Turtle [23] e RoboZZle [24]. Eles estabelecem um perfil comum de jogos deste tipo, que em geral tem como público-alvo crianças a partir de 5 anos e consiste em séries de problemas do tipo

“colete todos os objetos da fase”. Os jogadores devem utilizar blocos de comando para controlar o personagem e ajudá-lo a cumprir seu objetivo. A maioria destes jogos ensina funções, laços de repetição e alguns também incluem o uso de condições. Dos jogos listados, nenhum tem objetivo pedagógico compatível com a definição de Pensamento computacional concorrente, o objetivo comum é introduzir algoritmos e lógica de programação sequencial.

Uma busca na loja online *Steam*, na categoria “programação” apresenta vários jogos com mecânicas relacionáveis ao PC Concorrente. Muitos dos que se encontram em destaque são da empresa *Zachtronics*, cujos jogos que possuem classificação indicativa são classificados como *T-rated* [25], apropriados para adolescentes de 13 anos ou mais [26]. De acordo com a empresa, o motivo é o uso de linguagem inapropriado, referências a drogas, sexo e violência [25], o que os torna obviamente inadequados como ferramenta para o ensino de PC Concorrente para crianças de idade entre 4 e 10 anos, o público alvo do jogo desenvolvido nesse trabalho. A loja da *Steam* também mostra jogos com abordagem diferente, como por exemplo “7 Billion Humans” [27], que traz uma proposta que une um visual cartunesco e o controle de muitos personagens ao mesmo tempo, com um único código. Mesmo aparentando ter um público-alvo mais jovem, a dificuldade do jogo aumenta rapidamente, fazendo com os problemas alcancem um nível muito alto de complexidade, extrapolando o objetivo educacional para crianças.

B. Aplicativos

Saindo do escopo dos jogos, existem também aplicativos que visam introduzir jovens ao PC, como o Scratch [28], direcionado para crianças de idade entre 8 e 10 anos, e o Scratch Jr. (Sjr) [29], uma versão para crianças entre 5 e 7 anos. Nestes aplicativos, os usuários utilizam blocos de comando para criar sequências que definam o comportamento de personagens.

Em relação ao PC Concorrente, podemos notar compatibilidade nas duas ferramentas, pois cada personagem possui uma sequência própria de comandos que é executada de maneira sobreposta. O ponto diferencial do Scratch é que ele possibilita interação entre os personagens, diferente do Sjr. Desta forma, o ensino de PC Concorrente no Scratch se apresenta menos limitado, permitindo inclusive seu uso para criação de ementas com noções de computação concorrente para crianças, como feito na Grécia [30].

V. RETÓRICA PROCEDURAL COMO FERRAMENTA DE ANÁLISE

Utilizando a teoria de retórica procedural [6] como base, outros autores se propuseram a fazer análises de jogos. Por exemplo, Wandler faz uma abordagem narrativa sobre Portal e Portal 2 [31], e Harper explora as mecânicas do jogo Persona 3 [32].

Harper analisa três sistemas do jogo Persona 3 isoladamente. Seu objetivo na pesquisa foi enumerar os processos e descrever

o que estava sendo representado [32]. Como exemplo de análise feita, podemos citar a seguinte: o jogador tem a possibilidade de interpretar um papel escolhendo opções de diálogo e se relacionando desta maneira com seus amigos da história. Em dado momento, algum desses amigos pode solicitar um conselho, como, por exemplo, se deve fugir de casa a fim de chamar a atenção dos pais divorciados. Se o jogador encorajar o amigo a sair de casa, este personagem o verá como um *bom amigo* e prestará auxílio nos combates tornado-os mais fáceis. Caso contrário, esse personagem passa a ajudar menos em combates. Este incentivo, de tornar o combate mais fácil, para quem tomar a decisão imprudente pode fazer com que jogadores que estavam empenhados em construir para si um personagem coeso desistam da imersão em busca de um jogo mais fácil, diminuindo a qualidade da experiência.

VI. MAPA DO TESOURO: ENSINO DE PC CONCORRENTE

Como visto na seção anterior, a quantidade de *software* de ensino de PC Concorrente para crianças entre 4 e 10 anos é limitada. O jogo Mapa do Tesouro foi idealizado e desenvolvido pelos autores com o intuito ampliar a oferta de ferramentas para esse fim.

Esta seção é dividida em três partes: a introdução ao tema e as mecânicas principais do jogo; análise dos processos utilizando à Retórica Procedural; e por fim detalhamento dos elementos do jogo.

A. Visão Geral

A temática escolhida para o artefato foi a pirataria, com o uso de uma paleta de cores viva. A decisão foi tomada com a intenção de tornar o jogo atraente ao público infantil. O jogo é dividido em várias ilhas, cada uma representando uma fase e tem como objetivo encontrar um ou mais tesouros. Os tesouros podem ser baús, obrigatórios para completar a fase, ou joias que são opcionais.

O jogador controla um ou dois piratas, limitados a andar em um mapa quadriculado, como podemos ver na Fig. 1. Os piratas são programados por uma sequência de passos simples, como andar ou cavar. Uma ação de levantar bandeira permite sincronizar os piratas.



Fig. 1. Exemplo de fase do jogo.

Para movimentar os personagens, o jogador deve primeiro criar uma sequência de comandos a serem executados. Cada comando é representado como uma peça de um quebra-cabeça, o que indica que devem ser encaixados um a seguir do outro.

Nas fases iniciais, o jogador controla apenas um pirata e vai sendo apresentado a problemas que exigem caminhos, e programas, cada vez mais complexos. Mais tarde, nas fases de programação concorrente, o jogador controla dois piratas, cada um com uma sequência de comandos própria, ou seja, cada um executando uma tarefa própria. Ao clicar em executar, o jogo decide aleatoriamente, a cada iteração, qual pirata terá seu movimento executado. Algumas fases obrigam os dois personagens a se ajudar e a interagir.

Começamos o jogo com fases de execução linear (apenas um personagem) para garantir que qualquer criança, até aquelas que não tenham aprendido conceitos básicos de algoritmos, consigam superar os desafios e atingir as fases que apresentam problemas de natureza concorrente (com mais de um personagem).

O jogo foi idealizado e desenvolvido para permitir tanto a utilização individual e independente quanto a utilização guiada em sala de aula.

B. Articulando Elementos do Jogo e o Pensamento Computacional Concorrente

Na seção II definimos Retórica Procedural e sua relevância no contexto de jogos educacionais para dar suporte ao *game design*. Sob sua influência, os elementos e processos do Mapa do Tesouro foram concebidos considerando seu alinhamento com o propósito educacional do jogo.

Ao criar uma sequência de blocos, controlar até dois personagens, e buscar por tesouros o jogador atua sobre o jogo e, ativamente, contribui para seu protagonismo no processo de aprendizagem.

Abaixo detalhamos como os principais processos do jogo se relacionam com PC e o PC Concorrente.

Criar sequencia de blocos Ao sequenciar blocos, ou seja, criar uma ordem de comandos, o jogador está planejando suas ações. Pensar antes de agir é importante no contexto do Pensamento Computacional (PC).

Controlando um personagem Para dar comandos para um personagem, o jogador deve entender o contexto ao redor, considerando o posicionamento do personagem, obstáculos e o objetivo. Do ponto de vista do PC, também é essencial que se entenda o contexto do que está sendo manipulado, seja um personagem, um banco de dados ou um registrador.

Encontrando tesouros Encontrar os tesouros é o mesmo que solucionar um problema. Nem sempre o acesso pelo mapa é fácil e muitas vezes é preciso de criatividade. Este processo, em conjunto com o uso de comandos, é a aplicação padrão de algoritmos, pois é necessário que o jogador utilize uma sequência de comandos para achar a solução.

Controlar dois personagens Ao controlar dois personagens, o jogador deve entender dois contextos de posição, obstáculos

e objetivos diferentes. Deve pensar se os personagens vão interagir e como vão interagir. Como a ordem de execução deles não é previsível, é definida aleatoriamente, então o jogador precisa aprender a sincronizá-los usando o bloco da bandeira. Este processo é o que mais se relaciona com o PC Concorrente.

C. Detalhamento de Elementos do Jogo

Os processos apresentados pelo jogo foram materializados por elementos alinhados a proposta da estética visual e organizados em fases de complexidade e dificuldade crescentes. Apresentamos a seguir uma descrição dos comandos, das fases e obstáculos presentes no jogo. A Tabela I e a Tabela II apresentam representações pictóricas dos comandos e obstáculos, respectivamente.

1) *Comandos*: Os comandos que podem ser escolhidos pelo jogador, cada um representando uma ação simples dos personagens, são apresentados na Tabela I. Escolhemos comandos simples e relacionados ao tema do jogo, para que o público-alvo entendesse facilmente.

Uma de nossas decisões foi não utilizar blocos de repetição ou condicionais, pois nosso objetivo era ampliar o público-alvo para que as fases concorrentes pudessem ser desfrutadas até pelos mais novos. Acreditamos que se adicionássemos os blocos mencionados os desafios concorrentes poderiam ficar complexos.

Tabela I
COMANDOS

Nome do Comando	Descrição
 Olhar	Existem quatro direções para o personagem olhar, cada uma representada por um comando. São eles: olhar para cima, olhar para baixo, olhar para esquerda e olhar para direita.
 Andar	O personagem anda na direção que estiver olhando. Se o personagem esbarrar em obstáculos fixos, ou tentar andar sobre a água, o jogo emite uma mensagem de erro.
 Atirar	O projétil disparado pelo estilingue viaja na direção que o personagem estiver olhando. A munição consegue ativar canhões.
 Esperar um turno	O personagem espera um turno. Se estiver sobre uma jangada, o pirata se move junto com ela.
 Cavar	O personagem usa sua pá para cavar o ponto em que está. Se nenhum tesouro for encontrado o jogo emite uma mensagem de erro.
 Levantar bandeira	Disponível apenas em fases concorrentes. Ao levantar a bandeira, o pirata espera, sem agir, até que seu colega também levante a bandeira. Quando as duas bandeiras estiverem levantadas, os piratas abaixam-nas ao mesmo tempo e a execução continua normalmente.

2) *Fases e Obstáculos*: Cada fase é composta por um mapa quadriculado de duas dimensões, formado por:

- um conjunto de blocos de chão, em que o personagem pode andar;
- um conjunto de blocos de água, em que o jogador só consegue de mover se estiver em uma jangada; e
- um conjunto de obstáculos, apresentados na Tabela 2.

Tabela II
OBSTÁCULOS

Obstáculo	Descrição
 Caixa de Madeira	A caixa de madeira é um objeto não ultrapassável, fixo no chão e destrutível com canhões.
 Barril	Barril é um objeto que pode ser empurrado se o jogador o fizer no sentido correto. Pode ser destruído por canhões e vira uma jangada se empurrado sobre a água. Existem dois tipos de barril: vertical e horizontal. A diferença é no eixo que pode ser empurrado.
 Canhão	Se for acertado pelo Estilingue, o canhão atira para onde estiver virado. A bala de canhão destrói os objetos de madeira que obstruem o caminho (Barril e Caixa). Existem quatro tipos de canhão, apontando para cima, baixo, esquerda e direita.
 Pedras	Objetos fixos no chão que não podem ser destruídos nem ultrapassados.
 Jangadas	Ela se move em linha reta pela água, e fica imóvel até que um pirata fique em cima dela e use o comando "Esperar um turno". A cada espera, a jangada desloca-se uma unidade para onde estiver apontando, independente da direção do personagem. A jangada pode estar se movendo na horizontal ou na vertical. Se ela chega no final de seu caminho, seu sentido é invertido e ela continua se movendo.

VII. MÉTODO DE DESENVOLVIMENTO

Para o desenvolvimento do jogo, foi escolhida a ferramenta Unity [33] como *game engine*. O principal motivo foi a grande quantidade de plataformas disponíveis para publicar. Dessa forma seria mais fácil ampliar o acesso ao jogo. Outro fator importante foi a familiaridade dos pesquisadores com a ferramenta.

O desenvolvimento foi dividido em várias etapas (Fig. 2), mas podemos agrupá-las em três partes, cada uma com um produto intermediário que poderia ser testado e avaliado.

A. Idealização e prototipagem

A primeira parte durou cerca de dois meses. O passo inicial foi definir o propósito do jogo. A partir dele e com base na ferramenta Endo-GDC [34] (um *canvas* que auxilia a idealização de jogos com propósito) foi feito um *brainstorm* para definir os conceitos do jogo.

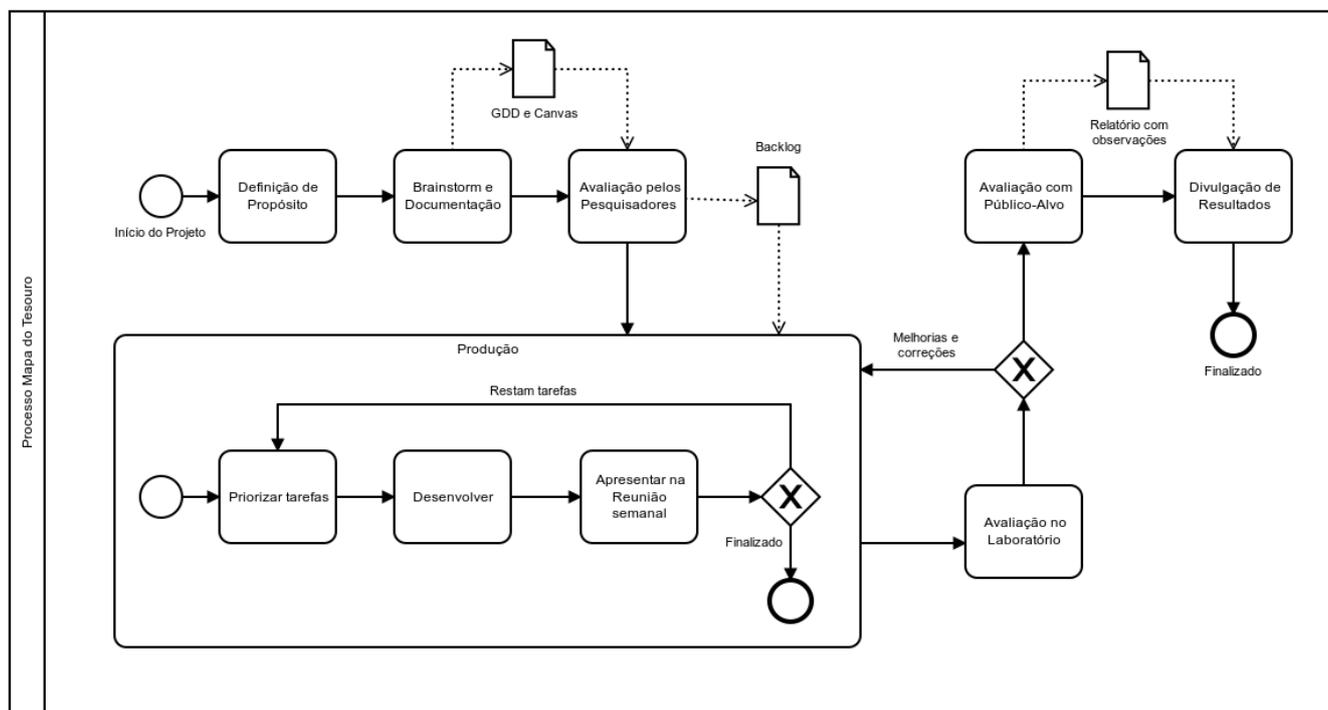


Fig. 2. Processo de desenvolvimento do Mapa do Tesouro

A partir da ideia formada, foi elaborado um *Game Design Document* (GDD) que deu sustentação para a programação de um protótipo simples para testes. Neste momento, o jogo não possuía elementos gráficos, apenas texto que aparecia em um terminal simulado. Nesta versão foi possível utilizar o jogo para verificar se as suas principais mecânicas, como definidas no GDD, inter-relacionavam-se de maneira adequada.

B. Produto Mínimo Viável e testes

O foco desta etapa foi a finalização de um produto mínimo viável (MVP) para avaliação, já com todas as mecânicas pretendidas para o jogo. O desenvolvimento ocorreu em ciclos semanais, seguindo a metodologia Scrum [35]. Cada semana o estado do jogo era apresentado nas reuniões do laboratório. Ao final do processo, o artefato apresentava a representação visual do mapa, objetos e também do personagem, além de possuir cinco fases.

A partir desta versão, foi feito um teste por alguns membros do laboratório. O grupo totalizou sete membros, que jogaram as fases e emitiram suas opiniões sobre o jogo.

O principal *feedback* recebido foi sobre uma falha do jogo ao adicionar comandos na lista usando um dispositivo com tela sensível ao toque, que prejudicaria usuários de *tablets* e celulares.

Com isso, o jogo voltou para produção, com meta de aperfeiçoar a interface visual do jogo. A inspiração escolhida foi o aplicativo Scratch Jr [29].

C. Finalização e testes com público-alvo

Por fim, na terceira e última etapa, foi criada uma versão final do jogo com quatorze fases, com foco em testes de qualidade com o público-alvo. O objetivo desta avaliação foi analisar o artefato em três aspectos:

- Dificuldade encontrada ao jogar
- Disposição de repetir a experiência
- Impressões sobre a diversão experimentada

Nesta avaliação, 39 crianças fizeram parte de 6 testes no total, um para cada faixa etária, que variou entre 4 e 9 anos de idade. É importante destacar que todas já possuíam cerca de 15 horas de prática extracurricular em um curso semanal de PC.

Os testes tiveram duração entre 30 e 50 minutos, variando de acordo com a disponibilidade de horário da turma. Com exceção das crianças de 4 anos, cada uma jogou individualmente em um *tablet*. Este último grupo jogou a maior parte do tempo com apenas um *tablet*, segurado pelo professor.

Durante os testes, o pesquisador responsável observou e documentou os comentários que surgiam e as dificuldades encontradas pelas crianças.

D. Análise de Resultados

As observações dos testes foram divididas em três grupos guiado pela convergência das observações, apresentadas na Tabela 3.

Todas as crianças se mostraram interessadas pela temática pirata, algumas até explicitamente dizendo que acharam o personagem “fofo”. Acreditamos que parte da facilidade de



Fig. 3. Aplicação do jogo **Mapa do Tesouro** em aula

Tabela III
GRUPOS DE TESTE

Grupo	Idade	Qnt.	Observações
A	4 anos	5	A melhor abordagem para esta idade foi manter o professor gerenciando o jogo e pedindo sugestões de movimentos para os alunos enquanto os auxiliava em qualquer dificuldade. As crianças apresentaram dificuldade com a diferença entre os blocos de Andar e Olhar (no jogo é preciso olhar na direção desejada antes de andar).
B	5 e 6 anos	9	Os alunos apresentaram dificuldade em entender o tutorial do início do jogo, que se revelou muito dependente do texto, foi necessária uma demonstração feita pelo professor. Ficou patente a necessidade de algumas fases iniciais para que o jogador aprenda o funcionamento dos blocos de Andar e Olhar, já adicionadas na última versão do jogo, e uma adaptação do tutorial inicial.
C	7, 8 e 9 anos	25	As crianças já eram alfabetizadas, mesmo assim não se interessaram em ler as mensagens de tutorial. Entretanto, não foi necessário que o professor interviesse para auxiliar no entendimento do funcionamento do jogo, exceto nas fases mais avançadas, nas quais os desafios são maiores.

entender o objetivo do jogo foi consequência do tema escolhido. As crianças assimilaram rápido que o personagem pirata deveria andar para cima dos “X” no mapa e cavar para achar tesouros.

Uma confusão recorrente foi a dificuldade em diferenciar a função do bloco andar, que é sempre para a direção para a qual o pirata olha, com os blocos de olhar em uma direção, que executam apenas uma rotação.

Três perguntas simples, alinhadas com os aspectos de qualidade avaliados, de resposta sim ou não foram feitas para os participantes dos testes. A escolha binária foi determinada

pela idade das crianças no intuito de evitar dificuldades de interpretação de uma escala mais complexa como as de Likert.

- 1) “Gostou do Jogo?”
- 2) “Você achou difícil?”
- 3) “Jogaria novamente?”

O resultado pode ser conferido na tabela IV.

Tabela IV
RESULTADOS

Pergunta	Respostas			
	Sim		Não	
Gostou do Jogo?	39	100%	0	0%
Achou difícil?	19	48,7%	20	52,3%
Jogaria Novamente?	39	100%	0	0%

Finalizado o último ciclo de desenvolvimento, o jogo foi lançado na Google Play [36], e disponibilizado no GitHub [37], com acesso aberto e gratuito.

VIII. CONCLUSÃO

Este artigo apresenta duas principais contribuições: do ponto de vista educacional, a exposição do conceito de PC Concorrente e a importância do desenvolvimento desta habilidade; e do ponto de vista de *game design*, a descrição e explicitação do processo de criação de um jogo educacional utilizando a retórica procedural para construção dos argumentos que sustentam os objetivos de aprendizagem.

O Mapa do Tesouro é um passo inicial no propósito de prover jogos para ensinar PC Concorrente para crianças entre 4 e 10 anos. As avaliações iniciais sobre a usabilidade e engajamento do artefato são positivas. Contudo, a efetividade do jogo como ferramenta de ensino merece ser melhor avaliada em um trabalho futuro.

Sob aspecto de idealização de um jogo educacional, a utilização da retórica procedural foi uma ferramenta útil para garantir que os processos apresentados construíssem os argumentos que sustentam os seus objetivos de aprendizagem, além de não impactar negativamente os objetivos pedagógicos.

Por fim, a formalização do processo em um diagrama (Fig. 2) torna explícita a divisão do desenvolvimento em ciclos, que incluíram testes e avaliações. Esta estratégia possibilitou que fossem detectados e solucionados problemas nas etapas iniciais de projeto, além de permitir o aprimoramento a partir das avaliações recebidas.

Esperamos que a apresentação do desenvolvimento deste trabalho encoraje e auxilie outras aplicações de estratégias semelhantes, contribuindo para construção de uma metodologia que facilite a criação de jogos educacionais, assegure o atingimento de seus objetivos de aprendizagem e mantenha seu caráter lúdico.

AGRADECIMENTOS

Este estudo foi parcialmente financiado pelo Conselho Nacional de Pesquisa (CNPq) - Brasil - RESOLUÇÃO NORMATIVA RN-017/2006 e pela Coordenação de Aperfeiçoamento

de Pessoal de Nível Superior (CAPES) - Brasil - Código de Finanças 001.

REFERÊNCIAS

- [1] S. Grover and R. Pea, “Computational thinking in k–12: A review of the state of the field,” *Educational researcher*, vol. 42, no. 1, pp. 38–43, 2013.
- [2] J. M. Wing, “Computational thinking,” *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.
- [3] “National curriculum in England: computing programmes of study.” [Online]. Available: <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>
- [4] R. Van Eck, “Digital game-based learning: It’s not just the digital natives who are restless,” *EDUCAUSE review*, vol. 41, no. 2, p. 16, 2006.
- [5] R. S. Monclar, M. A. Silva, and G. Xexéo, “Jogos com propósito para o ensino de programação,” in *Anais do Simpósio Brasileiro de Games e Entretenimento Digital - SBGames*, 2018.
- [6] I. Bogost, *Persuasive games: The expressive power of videogames*. MIT Press, 2007.
- [7] R. Hunicke, M. LeBlanc, and R. Zubek, “Mda: A formal approach to game design and game research,” in *Proceedings of the AAAI Workshop on Challenges in Game AI*, vol. 4, no. 1. San Jose, CA, 2004, p. 1722.
- [8] “BBC Bitesize.” [Online]. Available: <https://www.bbc.com/bitesize/topics/z7tp34j>
- [9] “Code.Org.” [Online]. Available: <https://code.org/curriculum/course3/1/Teacher>
- [10] “Google For Education.” [Online]. Available: <https://edu.google.com/resources/programs/exploring-computational-thinking/>
- [11] K. Kirkpatrick, “Parallel computational thinking,” *Communications of the ACM*, vol. 60, pp. 17–19, 11 2017.
- [12] I. Bogost, “The rhetoric of video games,” *The ecology of games: Connecting youth, games, and learning*, pp. 117–140, 2008.
- [13] B. Fishman, M. Riconscente, R. Snider, T. Tsai, and J. Plass, “Empowering educators: Supporting student progress in the classroom with digital games,” Ann Arbor: University of Michigan, Tech. Rep., 2014.
- [14] M. Prensky, “Digital game-based learning,” *Computers in Entertainment (CIE)*, vol. 1, no. 1, pp. 21–21, 2003.
- [15] K. Becker, “Pedagogy in commercial video games,” in *Games and Simulations in Online Learning: Research and Development Frameworks*. Idea Group, 01 2007, pp. 21–47.
- [16] R. Koster, *Theory of fun for game design*. O’Reilly Media, Inc., 2013.
- [17] C. Campbell, “The oregon trail was made in just two weeks,” Jul 2013. [Online]. Available: <https://www.polygon.com/2013/7/31/4575810/the-oregon-trail-was-made-in-just-two-weeks>
- [18] C. A. Maziero, “Sistemas operacionais: Conceitos e mecanismos,” *Livro aberto. Acessível em: http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php*, p. 27, 2014.
- [19] P. Pacheco, *An introduction to parallel programming*. Elsevier, 2011.
- [20] C. Breshears, *The art of concurrency: A thread monkey’s guide to writing parallel applications*. O’Reilly Media, Inc., 2009.
- [21] L. Inc., “Lightbot Hour.” [Online]. Available: <http://lightbot.com/hour-of-code.html>
- [22] H. WAHBA, “Cato’s Hike,” 1999. [Online]. Available: <http://hwahba.com/catoshike/>
- [23] N. is Great, “Move the Turtle,” 1999. [Online]. Available: <http://movetheturtle.com/>
- [24] I. OSTROVSKY, “RoboZZle,” 1999. [Online]. Available: <http://robuzzle.com/>
- [25] 2019. [Online]. Available: <http://www.zachtronics.com/zachademics/>
- [26] 2020. [Online]. Available: <https://www.esrb.org/ratings-guide/>
- [27] “7 Billion Humans,” 2018. [Online]. Available: <https://tomorrowcorporation.com/7billionhumans>
- [28] “Scratch.” [Online]. Available: <https://scratch.mit.edu/>
- [29] M. LLC, “Scratch Jr.” [Online]. Available: <https://www.scratchjr.org/>
- [30] E. Fatourou, N. C. Zygouris, T. Loukopoulos, and G. I. Stamoulis, “Teaching concurrent programming concepts using scratch in primary school: Methodology and evaluation,” *International Journal of Engineering Pedagogy (IJEP)*, vol. 8, no. 4, pp. 89–105, 2018.
- [31] Z. R. Wendler, “‘who am i?’ rhetoric and narrative identity in the portal series,” *Games and Culture*, vol. 9, no. 5, pp. 351–367, 2014.
- [32] T. Harper, “Rules, rhetoric, and genre: Procedural rhetoric in persona 3,” *Games and Culture*, vol. 6, no. 5, pp. 395–413, 2011.
- [33] Unity, “Unity 3D.” [Online]. Available: <https://unity.com/pt>
- [34] B. B. Taucei, “Endo-gdc: Desenvolvimento de um game design canvas para concepção de jogos educativos endógenos,” Master’s thesis, UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, Rio de Janeiro, 2019.
- [35] R. Sabbagh, *Scrum: Gestão ágil para projetos de sucesso*. Editora Casa do Código, 2014.
- [36] LUDES, “Mapa do Tesouro.” [Online]. Available: <https://play.google.com/store/apps/details?id=com.LUDES.MapaTesouro>
- [37] —, “Mapa do Tesouro.” [Online]. Available: <https://github.com/LUDES-PESC/PIBIC-JogosEducacionais>