

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

HORÁCIO BRESCIA MACÊDO HENRIQUES
JÚLIO RAMA KRSNA MANDOJU

LEO E MAYA: Um jogo para auxiliar no ensino de pensamento computacional para
crianças e adolescentes

RIO DE JANEIRO

2021

HORÁCIO BRESCIA MACÊDO HENRIQUES

JÚLIO RAMA KRSNA MANDOJU

LEO E MAYA: Um jogo para auxiliar no ensino de pensamento computacional para crianças e adolescentes

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profa. Carla Amor Divino Moreira Delgado

Co-orientador: Prof. Geraldo Bonorino Xexéo

RIO DE JANEIRO

2021

H5191

Henriques, Horácio Brescia Macêdo

LEO e Maya: um jogo para auxiliar no ensino de pensamento computacional para crianças e adolescentes / Horácio Brescia Macêdo Henriques, Júlio Rama Krsna Mandoju. – 2021.

91 f.

Orientadora: Carla Amor Divino Moreira Delgado.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Bacharel em Ciência da Computação, 2021.

1. Pensamento computacional. 2. Pensamento computacional paralelo. 3. Jogos educativos. I. Mandoju, Júlio Rama Krsna. II. Delgado, Carla Amor Divino Moreira (Orient.). III. Universidade Federal do Rio de Janeiro, Instituto de Matemática. IV. Título.

HORÁCIO BRESCIA MACÊDO HENRIQUES
JÚLIO RAMA KRSNA MANDOJU

LÉO E MAYA: Um jogo para ajudar no ensino de pensamento computacional para crianças e adolescentes

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 05 de Março de 2021.

BANCA EXAMINADORA:



Carla Amor Divino Moreira Delgado, Orientadora, D.Sc (UFRJ)

participação por videoconferência

Geraldo Bonorino Xexéo, Co-Orientador, D.Sc (UFRJ)

participação por videoconferência

Adriana Santarosa Vivacqua, D.Sc (UFRJ)

participação por videoconferência

Eduardo Freitas Mangeli de Brito, M.Sc (UFRJ)

AGRADECIMENTOS

Nossos agradecimentos mais profundos aos nossos pais, em primeiro lugar, que nos educaram e tornaram possível a trajetória até aqui. Nossos agradecimentos vão também à nossa orientadora Carla Delgado e ao nosso co-orientador Geraldo Xexéo pela orientação por todo o decorrer do trabalho mesmo em condições adversas, ao Departamento de Ciência da Computação para permitir que este trabalho viesse a acontecer, aos professores que voluntariamente responderam à pesquisa conduzida para validação deste trabalho e aos nossos animais de estimação, os Leo e Maya verdadeiros.

RESUMO

Em um mundo com cada vez mais computadores, é imprescindível que as pessoas saibam resolver problemas usando computadores de forma eficaz. Para tal, é necessário introduzir pessoas ao pensamento computacional: um meio de resolver problemas que consiste em interpretar e dividir problemas grandes em problemas menores com soluções conhecidas e descrever essa solução na forma de um algoritmo. Vários currículos ao redor do mundo, incluindo a Base Nacional Comum Curricular do Brasil, preveem que os alunos tenham contato com pensamento computacional, seja nas matérias de matemática ou dedicadas à computação. Contudo, nem todo mundo é efetivamente exposto a esse conteúdo, o que faz com que o contato das pessoas com pensamento computacional só venha a acontecer posteriormente e com pouca profundidade, sem ter contato formal com aspectos importantes como a concorrência.

O jogo Leo & Maya é uma forma de ajudar no ensino de pensamento computacional para crianças de sete a onze anos, que é a faixa etária em que devem ser introduzidos a esse modo de pensar de acordo com estudos pedagógicos abordados neste trabalho. O jogo consiste em vinte e duas fases de complexidade crescente onde o jogador é levado a montar algoritmos usando um conjunto específico de instruções para completar objetivos específicos, por vezes fazendo uso de pensamento computacional paralelo para controlar mais de um personagem dentro de uma fase. Durante as fases, dois princípios de pensamento computacional são fortemente trabalhados: a concepção de algoritmos e a divisão de problemas em subproblemas.

De modo a testar o produto deste trabalho enquanto ferramenta para o auxílio de ensino de jovens em idade escolar, o jogo foi passado a onze professores de diversas áreas e faixas etárias. Estes mesmos professores deram seus pareceres através de um formulário. O jogo foi avaliado enquanto uma ferramenta didática e enquanto uma atividade lúdica prazerosa para os alunos. Apesar da validação ter ocorrido durante a pandemia do Covid-19, que limitou o acesso aos professores entrevistados, a recepção por parte dos professores dentro da faixa etária pretendida foi positiva, apontando para um futuro promissor para o projeto.

Palavras-chave: Pensamento computacional. Pensamento computacional paralelo.

Jogos educativos.

ABSTRACT

In a world with more and more computers, it is essential that people know how to solve problems using computers efficiently. For such, it is necessary to introduce people to computational thinking: a way of solving problems that consists in interpreting and dividing large problems in smaller ones with known solutions and describing this solution in the form of an algorithm. Several curricula around the world, including the Base Nacional Comum Curricular (Common Core Curriculum) of Brazil, expect that students be exposed to computational thinking, be it in subjects such as mathematics or the ones dedicated to computing. However, not everyone is effectively exposed to this subject, which leads to people's first contact with computational thinking happening later in life and in a shallow way, missing important aspects such as concurrency.

The game Leo & Maya is a way to help the teaching of computational thinking to children ages seven through eleven, which is the intended age group for being introduced to this way of thinking according to pedagogic studies considered in this work. The game consists of twenty-two stages of increasing complexity where the player is asked to build algorithms using a specific set of instructions to complete specific objectives, sometimes making use of parallel computational thinking to control more than one character in a single stage. During the stages, two principles of computational thinking are strongly tackled: algorithms and decomposition.

In order to test the validity of the result of this work as a supporting tool for the teaching of youngsters in school age, the game was handed to eleven teachers of several subjects and age groups. These same teachers gave their opinions regarding the game through a form. The game was evaluated as a didactic tool and as a pleasant ludic experience for the students. Despite conducting the validation during the Covid-19 pandemic, which limited the access to the interviewed teachers, the reception from the teachers in the pretended age group was positive, pointing towards a promising future for the project.

Keywords: Computational thinking. Parallel computational thinking. Educational games.

LISTA DE ILUSTRAÇÕES

Figura 1: Exemplo de fase do jogo “Coding Planets”	27
Figura 2: Exemplo de fase do jogo “Mapa do Tesouro”	28
Figura 3: Exemplo de fase do jogo “Human Resource Machine”	29
Figura 5: Precedência entre conteúdos pedagógicos	37
Figura 6: Precedência entre mecânicas do jogo	40
Figura 7: Tela inicial da primeira fase do primeiro grupo de fases	43
Figura 8: Tela da interface de laços da primeira fase do quinto grupo de fases, perguntando por quantas vezes um laço de instruções específico deve se repetir	44
Figura 9: Tela de vitória da primeira fase do quinto grupo de fases	45
Figura 10: Tela de derrota da primeira fase do quinto grupo de fases, explicitando o motivo da derrota	45

LISTA DE GRÁFICOS

Gráfico 1: Relação de professores por faixa etária dos alunos	50
Gráfico 2: Resultados absolutos da pergunta “Este jogo é apropriado para a faixa etária dos seus alunos?”	61
Gráfico 3: Resultados absolutos da pergunta “Seus alunos gostarão de jogar este jogo?”	61
Gráfico 4: Resultados absolutos da pergunta “Você concorda que o jogo consegue ensinar os seguintes pilares de pensamento computacional?”	64

LISTA DE QUADROS

Quadro 1: Sugestões de atividades para o ensino de pensamento computacional	20
Quadro 2: Relação entre conteúdos pedagógicos e objetivos de aprendizado	36
Quadro 3: Relação entre mecânicas do jogo e objetivos de aprendizado	38
Quadro 4: Relação entre seção do jogo e conteúdo pedagógico	48
Quadro 5: Justificativas para a pergunta “Você considera jogos educativos bons recursos didáticos?”	52
Quadro 6: Relação entre faixa etária, matéria lecionada e opiniões sobre o jogo	60
Quadro 7: Relação entre faixa etária, matéria lecionada, até qual grupo de fases o voluntário jogou e respostas à pergunta “Você concorda que o jogo consegue ensinar os seguintes pilares de pensamento computacional?”	63
Quadro 8: Relação entre faixa etária dos alunos, matéria lecionada e respostas à pergunta “Como você acredita que este jogo ajudaria no aprendizado dos seus alunos?”	65
Quadro 9: Respostas da pergunta do formulário “Caso o jogo tenha algo que você desgostou, fale mais sobre, por favor”	67
Quadro 10: Respostas da pergunta do formulário “Se você pudesse mudar qualquer coisa nesse jogo, o que você mudaria?”	69

LISTA DE SIGLAS

PC – Pensamento Computacional

BNCC – Base Nacional Comum Curricular

PCP – Pensamento Computacional Paralelo

CHA – Conhecimento, Habilidade e Atitude

MDA – Mecânica, Dinâmica e Estética

ZDP – Zona de Desenvolvimento Proximal

SUMÁRIO

1 INTRODUÇÃO	13
1.1 OBJETIVO	14
1.2 METODOLOGIA	15
1.3 ORGANIZAÇÃO DO TEXTO	15
2 CONCEITOS BÁSICOS	17
2.1 JOGO	17
2.2 PENSAMENTO COMPUTACIONAL	19
2.2.1 Pensamento Computacional Paralelo	22
2.3 LINGUAGEM DE PROGRAMAÇÃO VISUAL	22
2.4 TEORIA DO DESENVOLVIMENTO COGNITIVO DE PIAGET	23
2.5 ZONA DE DESENVOLVIMENTO PROXIMAL DE VYGOTSKY	25
2.6 JOGOS SIMILARES	26
3 PROJETO DE JOGO EDUCATIVO PARA AUXÍLIO NO ENSINO DE PENSAMENTO COMPUTACIONAL	31
3.1 CONCEITO	31
3.1.1 Game Design Canvas para jogos educativos endógenos	31
3.1.2 Game Design Document	32
3.2 OBJETIVO PEDAGÓGICO	32
3.2.1 Conteúdo Pedagógico	33
3.2.2 Objetivos de Aprendizado	34
3.2.3 Relação entre objetivos de aprendizado e conteúdo pedagógico	36
3.3 O JOGO SEGUNDO O MODELO MDA	38
3.3.1 Mecânicas	38
3.3.2 Dinâmicas	41
3.3.3 Estética	41
3.4 ESTRUTURA DO JOGO	42
3.4.1 Estrutura de uma fase dentro do jogo	42
3.4.2 Planejamento de fases	46
4 VALIDAÇÃO DO PROJETO	49
4.1 DESCRIÇÃO DO CONTEXTO DA VALIDAÇÃO	49
4.2 PERFIL DOS VOLUNTÁRIOS	50
4.3 DESCRIÇÃO DA PESQUISA DE VALIDAÇÃO	53
4.3.1 Descrição da pesquisa	53
4.3.2 Descrição do formulário de pesquisa	54
4.3.3 Tratamento das respostas	56
4.4 RESULTADOS DA PESQUISA	58

4.4.1 Opinião acerca da aceitação do jogo por parte dos alunos	58
4.4.2 Pilares de pensamento computacional	62
4.4.3 Como o jogo ajuda no aprendizado?	65
4.5 CRÍTICAS E SUGESTÕES	67
5 CONCLUSÃO	71
5.1 TRABALHOS FUTUROS	72
REFERÊNCIAS	74
APÊNDICE A – GAME DESIGN CANVAS PARA JOGOS ENDÓGENOS (ENDO-GDC) DE LEO & MAYA	79
APÊNDICE B – DOCUMENTO DE GAME DESIGN (GDD) DO JOGO LEO & MAYA, ORGANIZADO CONFORME DESCRITO POR TAUCEI (2019)	80

1 INTRODUÇÃO

Pensamento Computacional (PC) é um assunto que tem ganhado cada vez mais relevância ao longo dos últimos anos em discussões sobre educação infantil (STEPHENSON, 2011). Citado pela primeira vez por Jeanette Wing em 2006, PC foi vagamente definido como “resolução de problemas, idealização de sistemas e compreensão do comportamento humano através de conceitos fundamentais à ciência da computação” (WING, 2006, pg. 33) e, desde então, se constitui em objeto de discussão no meio acadêmico, da definição até a aplicabilidade em salas de aula.

Hoje em dia, compreende-se que PC seja um conjunto de habilidades necessárias, comumente divididas em quatro técnicas (ou pilares): abstração, decomposição em subproblemas, reconhecimento de padrões e criação de algoritmos (BRACKMANN, 2016). Essas habilidades seriam úteis não só para um programador que precisa escrever códigos com algum objetivo prático, mas para qualquer pessoa que precise resolver um problema.

Levando em conta esse ponto, esse conjunto de habilidades envolve capacidades que não necessariamente são trabalhadas em sala de aula mais tradicionais, onde o saber é passado de forma bancária (FREIRE, 1996) entre a figura do professor que existe para expor informações e a figura do aluno que absorve as informações. Afinal, em uma sala de aula “bancária”, o professor pode ensinar a programar em uma linguagem ou a usar uma ferramenta, mas este modelo dificulta a formação de habilidades que compõem o que se entende por PC (DE FRANÇA et al, 2014). Portanto, em um cenário como este, o aluno teria de buscar esse conjunto de habilidades não apenas nos meios tradicionais de sala-de-aula, e com sorte os encontraria no jogo em questão, durante uma atividade lúdica em um momento de lazer.

No contexto deste trabalho, entende-se por habilidade a definição estabelecida pelo modelo CHA de caracterização de competências. Neste modelo, uma competência é caracterizada por três atributos: o conhecimento teórico de um conteúdo ou de um conjunto de conteúdos, a habilidade de utilizar os referidos conteúdos e a atitude para usar as próprias habilidades de modo a resolver problemas. Este modelo é muito utilizado em gestão de pessoas, mas neste contexto nós focaremos apenas na definição de habilidade, tendo em vista o objetivo do jogo de ensinar habilidades úteis aos seus jogadores.

Contudo, cada vez mais currículos de ensino médio e ensino fundamental estão agregando conteúdos de PC. Por exemplo, o plano de currículo K-12, dos Estados Unidos tem flertado com conceitos de PC desde 2011 (BARR e STEPHENSON), vindo a cada vez mais empregar esforços ao longo de anos para agregar conteúdos de PC no currículo de crianças e adolescentes. Já no Brasil, a Base Nacional Comum Curricular (BNCC) aprovada em dezembro de 2017 menciona explicitamente, na seção de ensino matemático, aplicações de álgebra no desenvolvimento de PC através do uso de algoritmos como estratégia para resolver problemas matemáticos (BNCC, 2018). Portanto, um jogo onde princípios de PC fossem ensinados com base nas experiências do jogador em um ambiente que “provoca a atividade” de pensar computacionalmente (NOVAESCOLA.ORG.BR, 2008) seria bem-vindo sob um ponto de vista pedagógico.

Outro detalhe de PC que currículos normalmente não abordam é pensamento computacional paralelo (PCP). Por mais que tenha sido mencionado por Wing em 2006 em seu primeiro artigo seminal sobre o assunto, e mais uma vez destrinchado em maior detalhe por Kirkpatrick (2017), PCP é tido não só como um assunto que pode ser complexo demais para um aluno ter tempo de aprender em sala de aula, apesar de sua importância (KIRKPATRICK, 2017). Logo, seria cômodo abordarmos essa faceta de PC que normalmente é ignorada ou apenas mencionada ao final de cursos que se propõem a lidar com o conteúdo em questão.

1.1 OBJETIVO

De modo a auxiliar o ensino de PC para jogadores de 7 (sete) a 12 (anos), este trabalho explicará o funcionamento do jogo educativo Leo & Maya, desenvolvido pelos alunos Horácio Macêdo e Júlio Rama do curso de Bacharelado de Ciência da Computação. O objetivo do jogo é ensinar, através de uma abordagem lúdica, como fazer uso de PC para a resolução de problemas que admitem soluções algorítmicas. Consideraremos o objetivo atingido caso o jogador seja capaz de jogar todas as fases do jogo voltadas a testar a compreensão acerca dos conceitos ensinados.

1.2 METODOLOGIA

De modo a atingirmos o objetivo proposto, o trabalho se constituirá em um jogo eletrônico voltado para o público de 7 (sete) a 12 (doze) anos dividido em fases responsáveis por ensinar conteúdos relacionados aos pilares de PC de modo que fosse educativo para jovens. Para desenvolver o jogo, nós nos organizamos de acordo com as diretrizes propostas pelo artigo do Endo-GDC (TAUCEI, 2019) de modo a conciliar o aspecto didático com o aspecto lúdico do produto final.

De modo a validar o projeto, levamos o protótipo para professores, com foco naqueles de escolas do Ensino Fundamental, de modo a coletar seus pareceres acerca do jogo, seu conteúdo e sua forma de se comunicar com os jogadores. Por causa da pandemia de Covid-19 e questões morais envolvendo experimentação com menores de idade, não tivemos a possibilidade de entrevistar diretamente o nosso público-alvo, o que nos levou a ir atrás das figuras mais próximas do nosso público-alvo.

As opiniões de professores foram coletadas através de um questionário através da ferramenta *Google Forms*. No referido questionário, nós damos uma breve explicação textual sobre PC, passamos o *link* para uma versão do jogo para navegadores *web* e pedimos para que o educador jogue antes de responder às perguntas. Por fim, perguntamos as opiniões dos professores com base na convivência com seus alunos e sua experiência no magistério.

A validação também se dá através da comparação a nível teórico com outros jogos que tenham sido utilizados em experimentos similares sobre uso de jogos para ensino de PC. A comparação é feita usando o modelo MDA para analisar os jogos e mostrar que o nosso projeto está alinhado com outras iniciativas que se mostraram eficazes no passado.

1.3 ORGANIZAÇÃO DO TEXTO

O restante deste trabalho está dividido em outros três capítulos. No capítulo 2, são apresentados alguns conceitos necessários para a compreensão deste trabalho, onde explicamos as escolhas tomadas tanto na concepção do projeto quanto no desenvolvimento do jogo. No capítulo 3, entramos em detalhes sobre o jogo em si, mostrando o planejamento feito por trás do jogo, elucidando como os conteúdos pedagógicos se relacionam com o conteúdo

do jogo. No capítulo 4, discutimos tanto os métodos de experimentação conduzidos quanto os resultados obtidos. Por fim, no capítulo 5, são feitas as considerações finais sobre o trabalho e deliberações sobre o futuro do projeto.

2 CONCEITOS BÁSICOS

Neste capítulo, apresentaremos os conceitos utilizados na concepção deste trabalho. Por se tratar de um jogo educativo, discutiremos a definição de jogo adotada na seção 2.1, e o conteúdo a ser ensinado na seção 2.2. A seção 2.3 será dedicada a falar sobre programação visual, que está por trás da principal interação do jogador com o jogo. Na seção 2.4, abordaremos como nós escolhemos a faixa etária para o nosso jogo. Na seção 2.5, trataremos sobre a filosofia por trás de como transmitir informação para o jogador. Por fim, na seção 2.6, analisaremos produtos similares que já existam no mercado, até como forma de validação dos nossos esforços e forma de aprendizado sobre o que já foi feito nesta área.

2.1 JOGO

Formalmente, não existe uma única definição precisa sobre o que é um jogo. (XEXÉO et al., 2017) Ao longo dos anos, várias definições de jogos foram cunhadas. A mais tradicional das definições é a definição de Johan Huizinga (1950, p. 13), na qual ele defende que jogos são “uma atividade voluntária conscientemente alheia da vida ‘ordinária’ ao ser ‘não-séria’, mas ao mesmo tempo absorvendo o jogador intensa e completamente.” Outras definições foram dadas por nomes como Roger Caillois em 1962, Chris Crawford em 1982 e Katie Salen e Eric Zimmerman em 2004, mas a definição de jogo que melhor serve ao propósito deste trabalho é a definição de Jesper Juul (2005, p. 36), na qual diz que “jogos são sistemas baseados em regras com resultados variáveis e quantificáveis, onde diferentes resultados possuem diferentes valores, o jogador se esforça de modo a influenciar o resultado, o jogador se torna emocionalmente engajado com o resultado e as consequências da atividade são negociáveis (com ou sem consequências na vida real)”.

Esta última definição nos serve melhor do que as outras definições mais tradicionais uma vez que ela defende que o jogo pode possuir consequências na vida real. Isso se dá por admitir a existência de jogos com objetivos além de uma atividade “alheia da vida ‘ordinária’ (...) sem interesse material” de onde nenhum lucro se tira, podendo ser usados como uma forma de educar o público. O termo que normalmente é usado para descrever esse nicho de

jogos se chama “serious games”, ou “jogos sérios”, de acordo com Sawyer (2002 apud DJAOUTI; ALVAREZ; JESSEL, 2011). A ideia de jogos sérios tem sua origem nos anos 70, no campo da educação, apesar de jogos sérios serem feitos hoje em dia por uma miríade de motivos, desde que o jogo ofereça algo além do mero entretenimento (DJAOUTI; ALVAREZ; JESSEL, 2011).

Jogos, principalmente os eletrônicos, são analisados sob um modelo denominado “mecânica, dinâmica e estética”, ou *MDA framework* (HUNICKE; LEBLANC; ZUBEK, 2001). Esse modelo é uma forma de compreender jogos eletrônicos, “fechando o vão entre game design e desenvolvimento, crítica de jogos e pesquisa técnica de jogos” de acordo com Hunicke, LeBlanc, Zubek (2001, p.1). Esses três aspectos são importantes de serem levados em consideração por todos os envolvidos no desenvolvimento de um jogo, por mais que cada parte tenha a sua especialidade, de modo a criar um produto coerente. Além disso, esses aspectos são igualmente importantes para os que procuram compreender uma obra de modo a estudá-la.

O modelo MDA decompõe um jogo como sendo um conjunto de regras que formam sistemas que, por sua vez, extraem alguma resposta emocional do jogador. Essas regras são chamadas de mecânicas, os sistemas formados pelas regras são chamados de dinâmicas e as respostas emocionais do jogador são chamadas de estética.

Jogos sérios voltados à educação, contudo, têm um detalhe ao qual se deve atentar. Halverson (2005) determina, ao longo de seu texto “*What can K-12 school leaders learn from video games*” a existência de duas categorias mutuamente excludentes: jogos exógenos e jogos endógenos. Jogos sérios voltados à educação exógenos seriam todos aqueles onde o conteúdo pedagógico não está diretamente ligado com o *game design* do jogo, podendo assim um ser extraído do outro sem nenhuma perda para ambas as partes (ou seja, o jogo poderia existir sem o material didático e vice-versa). Enquanto isso, jogos sérios voltados à educação endógenos são jogos onde o jogo e o conteúdo pedagógico existem de forma simbiótica, de modo que dominar uma das partes acaba levando a dominar a outra.

De modo a maximizar o potencial educativo do jogo, Leo & Maya foi concebido como um jogo sério endógeno. Desta forma, esperamos que o ato de jogar o jogo seja suficiente para que jogadores aprendam sobre PC.

2.2 PENSAMENTO COMPUTACIONAL

Pensamento computacional (PC) é mais do que meramente saber pensar para programar. PC envolve resolver subproblemas, desenhar sistemas e compreender comportamento humano ao se utilizar de conceitos fundamentais à ciência da computação (WING, 2006). É uma síntese de formas de pensar que são normalmente divididas em quatro pilares fundamentais: a decomposição de problemas, o reconhecimento de padrões, a abstração de informações e a formulação de algoritmos (BBC.CO.UK/BYTESIZE).

Essa forma de pensamento e resolução não é útil somente ao programador, mas também para qualquer pessoa. De acordo com Jeanette Wing (2010, p.2), PC influencia outros campos de estudo, como “medicina algorítmica, arqueologia computacional, economia computacional, finanças computacionais, computação e jornalismo, direito computacional, ciências sociais computacionais e humanidades digitais”. E mesmo fora destes campos de estudo, PC é útil uma vez que dá às pessoas a capacidade de adaptar soluções algorítmicas, avaliar quando um computador é melhor indicado para lidar com um problema e aplicar estratégias como divisão e conquista na resolução de problemas na vida cotidiana (WING, 2010).

PC é uma habilidade comumente relacionada com currículos de computação (K12.COM) ou matemática, uma vez que os alunos precisam, de acordo com Base Nacional Comum Curricular, “ser capazes de traduzir uma situação dada em outras linguagens, como transformar situações-problema, apresentadas em língua materna, em fórmulas, tabelas e gráficos e vice-versa” (2018, p.271) para a aprendizagem de álgebra e outros campos da matemática. A BNCC também faz menção aos outros pilares de PC, uma vez que “cumpre salientar a importância dos algoritmos e de seus fluxogramas, que podem ser objetos de estudo nas aulas de Matemática” (2018, p.271). Também se faz menção à decomposição em subproblemas, uma vez que a BNCC compara um algoritmo à “decomposição de um procedimento complexo em suas partes mais simples” (2018, p.271). Por fim, a BNCC explicitamente menciona o pilar da identificação de padrões “para se estabelecer generalizações, propriedades e algoritmos” (2018, p.271).

A discussão sobre o ensino de PC não é atual, tendo em vista que a implementação de medidas, como por exemplo o uso de LOGO ou de Basic para fins didáticos, são discutidas

desde os anos 80 (PAPERT, 1980 apud GROVER e PEA, 2012). Contudo, a atenção ao assunto cresceu quando, em 2006, Wing definiu em seu artigo “*Computational Thinking*” o que é PC e estabeleceu a importância dessa prática. Desde então, trabalhos têm sido produzidos de modo a explorar formas de integrar o ensino de PC em turmas de escola (BARR e STEPHENSON, 2011; DE FRANÇA et al, 2014; REPENNING; BASAWAPATNA; ESCHERLE, 2016; MONCLAR et al, 2018).

No artigo “*Bringing computational thinking to K-12*”, Valerie Barr e Chris Stephenson listaram propostas de formas como conceitos de PC podem ser explorados por professores de diferentes áreas. O artigo em questão lista propostas de acordo com cinco grandes áreas: computação, matemática, ciência, estudos sociais e linguagens para estimular diferentes aspectos do uso de CT, dos seus pilares até conceitos tangenciais. As propostas variam entre aprender a ler gráficos, aprender a usar ferramentas até mesmo escrever receitas e jogar jogos endógenos como *Age of Empires* como forma de aprender sobre história.

O quadro 1 abaixo corresponde a uma tradução livre da tabela presente no referido artigo:

Quadro 1: Sugestões de atividades para o ensino de pensamento computacional

Conceito/Pilar de PC	Computação	Matemática	Ciência	Estudos sociais	Linguagens
Coleta de Dados	Encontrar uma fonte de dados para uma “região problema”	Criar amostras de uma “região problema” fazendo exercícios de probabilidade, por exemplo, virando moedas ou jogando dados	Coletar dados de um experimento	Estudar estatísticas de batalha, ou dados populacionais	Fazer análise linguística de frases
Análise de Dados	Escrever um programa para fazer cálculos estatísticos simples com base em dados	Contar número de ocorrências de jogadas de moeda ou rolagem de dados e analisar os resultados	Analisar dados de um experimento	Identificar modas em dados através de estatísticas	Identificar padrões para diferentes tipos de frase
Análise e representação de dados	Usar estrutura de dados como <i>array</i> , lista encadeada, grafos, tabela <i>hash</i> , etc	Usar um histograma, gráfico de pizza, gráfico de barras, etc. para representar dados; usar conjuntos, listas, grafos etc. para armazenar dados	Resumir dados de um experimento	Resumir e apresentar as modas	Representar padrões de diferentes tipos de frase
Abstração	Usar procedimentos para encapsular um conjunto de comandos repetidos frequente-	Usar variáveis em álgebra; identificar fatos essenciais em um problema enunciado	Construir o modelo de uma entidade física	Resumir fatos; extrair conclusões de fatos	Usar sinônimos e metáforas

	mente que formam uma função				
Análise e validação de modelo	Validar um gerador aleatório de números	Ajuste de curvas	Validar que um modelo é correto		
Automação		Usar ferramentas como: <i>Geometer Sketchpad</i> ; <i>Star Logo</i> ; amostras de código <i>Python</i>	Usar softwares de testagem	Usar Excel	Usar um corretor ortográfico
Testagem e verificação	Depurar um programa; escrever testes unitários; verificação formal do programa	Tentativa-e-erro	Validar e tratar dados		
Algoritmos e procedimentos	Estudar algoritmos clássicos; implementar um algoritmo para uma “região problema”	Fazer fatoração de divisões extensa; usar “vai um” em operações de adição e subtração	Fazer um procedimento experimental		Escrever instruções
Decomposição de problemas	Definir objetos e métodos; definir função main e outras funções	Aplicar ordem de operações em uma expressão	Classificar espécies		Escrever uma escaleta
Controle de estruturas	Usar condicionais, laços, recursão, etc.	Estudar função em álgebra como se estudam funções em programação; usar iteração para resolver problemas enunciados em palavras		Escrever uma história não-linear	
Paralelização	<i>Threading</i> , <i>pipelining</i> , dividir dados ou tarefas de modo a serem processadas em paralelo	Resolver sistemas lineares; fazer multiplicação de matrizes	Rodar experimentos de forma simultânea com diferentes parâmetros		
Simulação	Animação de algoritmos, varredura de parâmetros	Representar uma função em um plano cartesiano e modificar os valores das variáveis	Simular o movimento de planetas no Sistema Solar	Jogar <i>Age of Empires</i> ; <i>Oregon Trail</i>	Encenar uma história

Fonte: Barr e Stephenson (2011, p. 117)

O jogo *Leo & Maya* tem algumas ideias do quadro acima incorporadas nas próprias mecânicas. O controle de estruturas e a paralelização no jogo são bem similares às tarefas que podem ser vistas em Computação segundo o quadro acima, visto que o jogador utiliza condicionais, laços e múltiplas *threads* — na forma de personagens — para resolver os desafios.

2.2.1 Pensamento Computacional Paralelo

Outra questão na mesma área é o pensamento computacional paralelo. Segundo “Parallel Computational Thinking” de Keith Kirkpatrick (2017), o pensamento computacional vem sendo estudado principalmente de forma sequencial, porém ele aponta diversas vantagens para o estudo do paralelismo. No mundo de hoje, pessoas têm acesso a computadores, smartphones e tablets com multiprocessadores, permitindo que todo tipo de aplicação execute tarefas em paralelo, algo que não era possível até aproximadamente uma década atrás, quando a computação paralela se tornou convencional.

Segundo Charles Weems (2017), no mesmo artigo, um dos benefícios da computação paralela é que ela “ajuda a desenvolver uma abordagem mais flexível para solucionar problemas porque há mais modelos nos quais se basear.” Além disso, o artigo menciona que o paralelismo oferece mais formas de quebrar um problema grande em subproblemas menores, algo que pode acrescentar às vantagens de se aprender o pensamento computacional, considerando que a decomposição de problemas é um de seus pilares fundamentais.

No artigo de Weems (2017) citado acima, podemos ver também um problema na aplicação atual dos conceitos de concorrência em PC nas salas de aula. No que se diz respeito a crianças, considera-se importante apenas o aprendizado de pensamento computacional sequencial. Um dos motivos para que se defenda essa baixa prioridade é a ênfase por parte dos currículos na parte sequencial, reconhecida logicamente como mais importante de ser aprendida antes de começar a lidar com concorrência. Contudo, algumas ferramentas já utilizadas por instituições de ensino que possuem aulas voltadas para PC, como por exemplo o Scratch (2007), já contam com algum grau de concorrência.

2.3 LINGUAGEM DE PROGRAMAÇÃO VISUAL

De acordo com Myers (1986, p. 60): “programação visual é qualquer sistema que permite ao usuário especificar um programa em duas (ou mais) dimensões (...) usando fluxogramas convencionais e linguagens de programação gráficas”. Por sua vez, a definição de programa usada por Myers é dada por Oxford (1983, p.) como “um conjunto de instruções

que podem ser submetidos como uma unidade para computadores de modo a direcionar o comportamento de sistemas.”.

No contexto deste trabalho, essas definições são importantes para validar a interação mecânica do jogador com o jogo. Ao jogar as fases e resolver os quebra-cabeças propostos, o jogador interage o tempo todo com uma interface de programação visual desenhada especificamente para o contexto deste jogo. A intenção por trás dessa decisão é democratizar o acesso do jogo a crianças e adolescentes em diferentes estágios de alfabetização e familiaridade com computação, anulando a necessidade de interpretar código escrito.

Linguagens de programação visual não são necessariamente mais simples de se ensinar do que linguagens de programação não visuais, tendo em vista que usar a parte gráfica de forma útil é, por si só, uma habilidade a ser aprendida (PETRE, 1995). Contudo, Petre (1995, pg. 41-43) também reconhece que abordagens gráficas aparentam ser mais atraentes para as pessoas de modo geral, admitindo que “a ilusão de acessibilidade pode ser mais importante do que a realidade.”, e conclui que abordagens gráficas que visam auxiliar usuários menos experientes devem mitigar ao máximo a preocupação do usuário no que diz respeito a como organizar o próprio código. Caso contrário, a abordagem gráfica provavelmente causaria mais confusão do que ajuda.

Apesar do sucesso de linguagens como Scratch (RESNICK, 2009) na educação de jovens, o nosso trabalho apresenta uma forma bastante restrita de programação visual, onde o jogador declara instruções em uma lista de tamanho e representação visual fixas. Essa estratégia não só mantém a consistência do jogo ao longo das fases, não exigindo que o jogador reaprenda a lidar com a interface, como também permite que o jogador possa focar em desenvolver o pensamento computacional.

2.4 TEORIA DO DESENVOLVIMENTO COGNITIVO DE PIAGET

Jean Piaget foi um psicólogo e epistemólogo influente no campo da educação, sendo uma das maiores referências existentes na literatura sobre pedagogia. Ao longo de uma extensa bibliografia, o aspecto mais importante para o nosso trabalho é a sua teoria de desenvolvimento cognitivo, descrita no livro *The Origins of Intelligence in Children* (1936).

Dentro da teoria de desenvolvimento cognitivo, a noção de estágios da formação da cognição nos foi particularmente útil. Este conceito defende que crianças de diferentes idades operam de formas distintas, de modo a serem capazes de apropriadamente aprender conteúdos de complexidade crescente ao longo de seu desenvolvimento. Enquanto que o livro supracitado defende a existência de quatro estágios dos primeiros meses de vida até o final da adolescência, nada garante que a criança passará pelos estágios no tempo previsto pela teoria, tendo em vista que o aprendizado é um processo que leva em conta tanto características biológicas de uma pessoa quanto estímulos externos. (MCLEOD, 2018).

A teoria do desenvolvimento cognitivo foi importante para determinarmos a faixa etária do público-alvo do jogo. Piaget (1936) descreveu quatro estágios: o estágio sensomotor, dos 18 até os 24 meses de idade, o estágio pré-operacional, dos dois até os sete anos de idade, o estágio operacional concreto, dos sete até os onze anos de idade, e por fim o estágio operacional formal, cobrindo o espaço de tempo da adolescência até a vida adulta. É importante destacar que Piaget nunca determinou as faixas etárias supracitadas como sendo fixas, mas como aproximações de quando uma criança atinge determinado estágio.

Ao definirmos a faixa etária alvo do nosso trabalho como sendo jovens de sete a onze anos de idade, nós procuramos atingir toda a faixa etária no estágio operacional concreto da formação da cognição. Neste estágio, a criança aprende de forma lógica e sistemática a manipular símbolos relacionados a objetos concretos (WOOD, SMITH e GROSSNIKLAUS, 2001). No caso, o jogador desejado já seria capaz de aprender a traçar relacionamentos entre instruções escolhidas e eventos que ocorrem como consequência, uma vez que este é o estágio onde o pensamento lógico começa a aflorar.

Essa teoria é muito utilizada por educadores ao defender uma educação voltada a fazer o aluno protagonista no próprio aprendizado. Como cada indivíduo se desenvolve à sua maneira, seria papel do educador provocar a busca por conhecimento. “Educar, para Piaget, é ‘provocar a atividade’” (FERRARI, 2008). Portanto, acreditamos que um jogo onde princípios de PC fossem ensinados com base nas experiências do jogador em um ambiente que “provoca a atividade” de pensar computacionalmente seria bem-vindo de um ponto de vista pedagógico.

2.5 ZONA DE DESENVOLVIMENTO PROXIMAL DE VYGOTSKY

Lev Vygotsky (1896-1934) foi um importante psicólogo bielorrusso que, assim como Piaget, estudou o desenvolvimento intelectual de crianças. Diferente do biólogo francês Piaget, que atribuía o desenvolvimento de conhecimento a processos internos que acontecem dentro da mente de um aluno, Vygotsky inspirou através de seu trabalho uma corrente chamada socioconstrutivismo, atribuindo às relações sociais papel predominante no aprendizado (FERRARI, 2008).

De todo o trabalho de Vygotsky, a Zona de Desenvolvimento Proximal (ZDP) foi a contribuição mais relevante no contexto deste trabalho. ZDP é a noção de que um aluno, dotado de algum conhecimento pregresso, é capaz de aprender conteúdos novos por si só a partir do que já se sabe (SARIKAS, 2020). Essa ideia é refletida na própria forma como o nosso jogo foi estruturado, uma vez que cada fase procura fazer o jogador utilizar o que já se sabe como base para resolver problemas propostos por fases futuras.

Normalmente, ao se falar sobre ZDP, muito se fala também sobre *scaffolding*, que seria o ato de ter o auxílio de um mentor na hora de adquirir algum aprendizado (SARIKAS, 2020). Essa pessoa mais experiente serviria como mentor para aqueles em processo de aprendizado, sendo normalmente a figura de um professor dentro de sala de aula, ou de um aluno mais velho que ajude os mais novos.

Reconhecer que *scaffolding* é uma parte de ZDP é importante no nosso caso, tendo em vista que não temos controle de quando os jogadores jogarão o nosso jogo. Ou seja, por mais que as fases tenham sido pensadas de modo a ensinar o jogador a pensar computacionalmente, nada garante que um jogador conseguirá usar do conhecimento utilizado em uma fase para, por si só, passar de fases futuras. Logo, nada impede um jogador menos experiente de recorrer a um jogador com mais experiência para cumprir um papel de mentor na hora de ensinar o necessário para se passar por fases, permitindo então que ocorra o *scaffolding*.

2.6 JOGOS SIMILARES

Este jogo não existe num vácuo, tendo em vista que existe todo um mercado de jogos sérios, tanto em computadores pessoais quanto em celulares, que se vendem com a proposta de serem educativos ao ensinar o jogador a programar. Esses jogos possuem diversas abordagens ao ensino de programação, tanto a nível de contexto quanto de execução. Os exemplares estudados serão explicados em maior detalhe a seguir.

Na plataforma Android, existe um vasto número de jogos muito parecidos, feitos normalmente por equipes pequenas e recaindo nas mesmas convenções. Todos os exemplos estudados por nós em celulares focavam na parte de montar algoritmos para completar desafios que normalmente envolvem mover personagens por caminhos, sendo os exemplos mais notáveis os jogos *Lightbot: Code Hour*¹, *Algorithm City*² e *Coding Planets*³. Todos esses jogos possuem uma abordagem bastante lúdica, *Lightbot* e *Algorithm City* sendo inclusive quase idênticos, e todos eles têm uma abordagem sem muito rigor no modo como introduzem ideias ao jogador, com a dificuldade variando de fase para fase sem apresentar uma progressão natural de dificuldade. Contudo, os três jogos são muito bem avaliados na PlayStore, e serviram de prova que esse gênero de jogo não só é possível, como agrada um certo nicho.

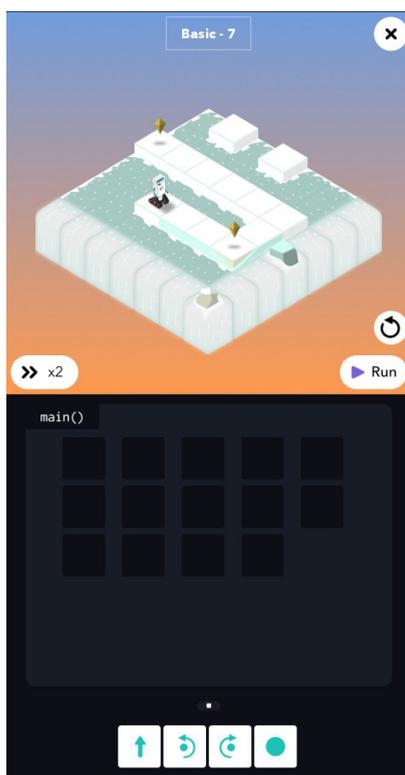
O paralelismo e os laços de *Leo & Maya* não são encontrados nestes jogos. Estes jogos abordam repetições de instruções apenas através de funções recursivas, sem exigir que o jogador elabore uma condição de parada para a recursão.

¹ **Lightbot: Code Hour** – Apps no Google Play. Disponível em: <https://play.google.com/store/apps/details?id=com.lightbot.lightbothoc>. Acesso em: 5 jun. 2020.

² **Algorithm City** – Apps no Google Play. Disponível em: <https://play.google.com/store/apps/details?id=air.MusterenGames.ElHarezmiCoding>. Acesso em: 5 jun. 2020.

³ **Coding Planets** – Apps no Google Play. Disponível em: https://play.google.com/store/apps/details?id=com.material.design.codingplanet&hl=pt_BR. Acesso em: 5 jun. 2020.

Figura 1: Exemplo de fase do jogo “Coding Planets”



Fonte: Acervo do Autor (2020)

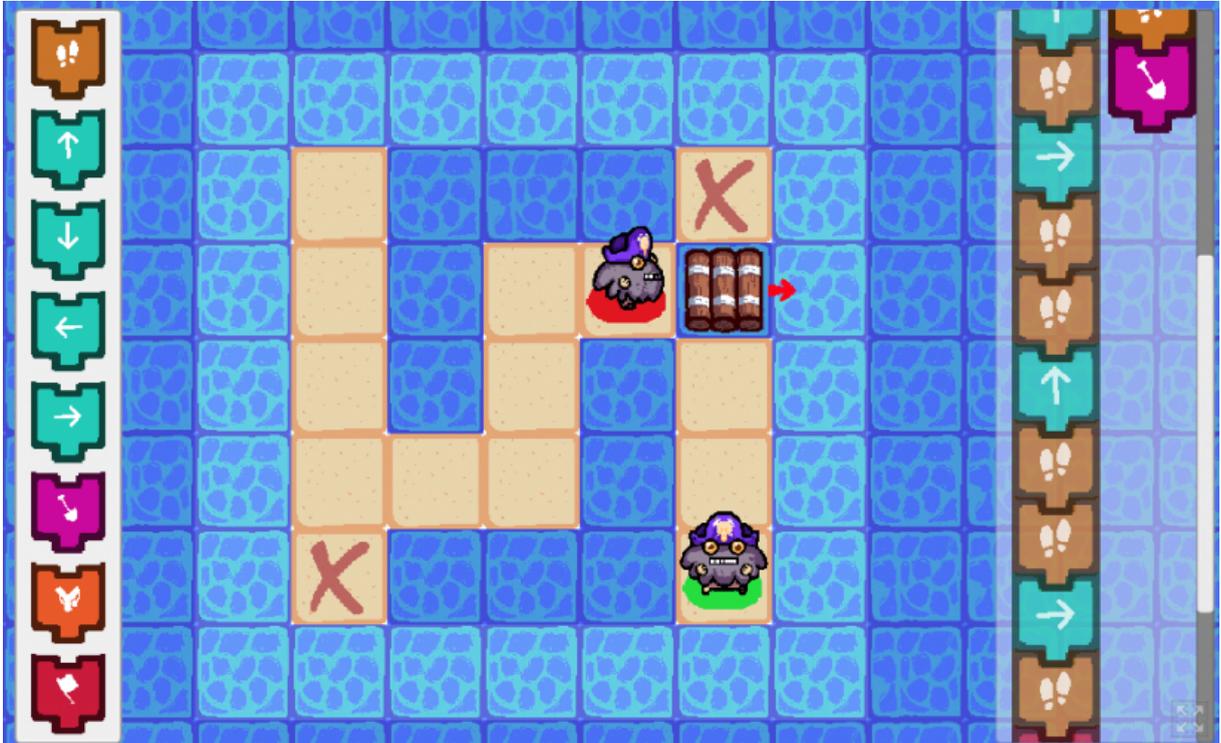
Outro jogo tomado como inspiração é o Mapa do Tesouro⁴, do aluno bolsista Pedro Marques (2020). Também desenvolvido para plataformas Android, este jogo possui um diferencial dos outros exemplares no que diz respeito à interface e às ideias que o jogo aborda. Enquanto que os jogos citados no parágrafo anterior se limitam a ensinar os mesmos conceitos na mesma ordem (como montar um algoritmo básico e como usar funções para repetir instruções indefinidamente através de recorrência), Mapa do Tesouro aborda noções de concorrência em suas fases, por vezes colocando o jogador em controle de mais de um personagem. Isso nos provou que não só a ideia é viável para um jogo infantil, como também é possível em um jogo para celular, onde o espaço para interface é menor.

Por mais que os dois jogos abordem concorrência, a forma como ela ocorre é diferente: em Mapa do Tesouro, os personagens se movem alternadamente de forma imprevisível, como *threads* compartilhando o mesmo processador; em Leo & Maya, os personagens se movem simultaneamente, como *threads* em processadores separados. O nosso

⁴ NASCIMENTO, Pedro Vitor Marques. **Mapa do Tesouro**. Disponível em: <https://tanookihouse.itch.io/mapa-do-tesouro>. Acesso em: 13 dez. 2020.

jogo também possui uma mecânica a mais, que é a definição de laços por parte do jogador, como forma de lidar com um espaço finito para declarar instruções.

Figura 2: Exemplo de fase do jogo “Mapa do Tesouro”

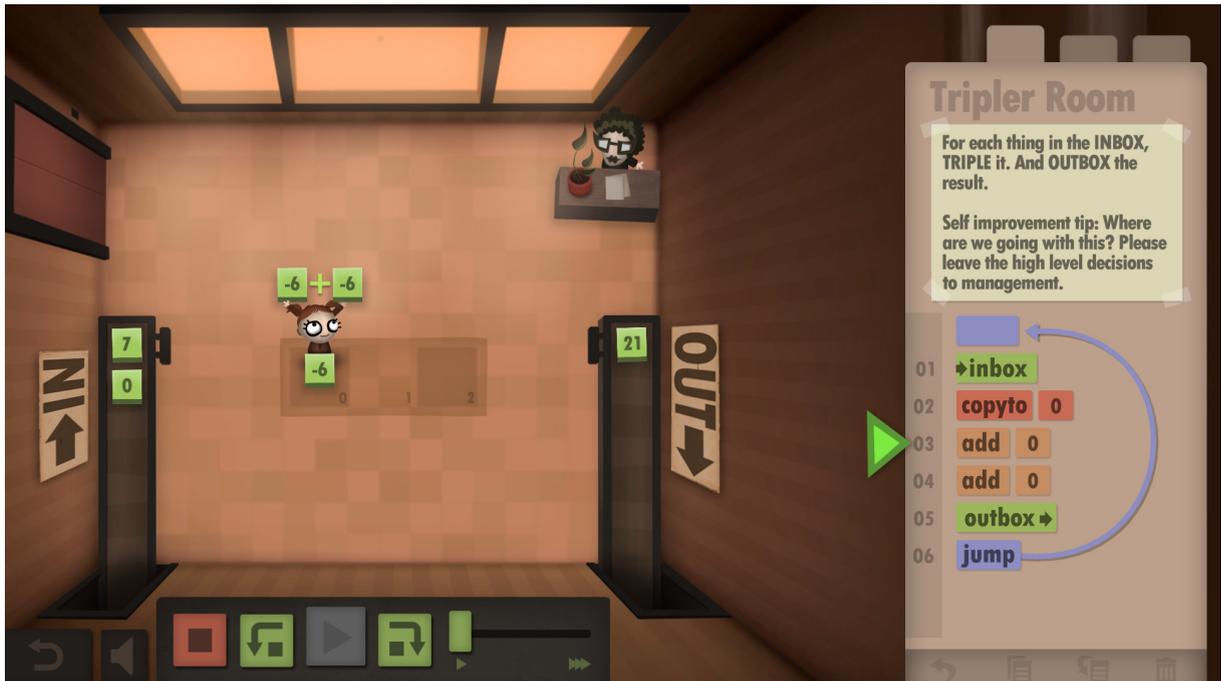


Fonte: Site do LUDÉS, acessado em 2020

Por fim, tomamos também como inspiração o jogo Human Resource Machine⁵ em como lidar com laços de instruções sem adotar a abordagem dos outros jogos que se dispuseram a tratar do assunto. Enquanto que, nos outros jogos, laços são abordados apenas através de recorrência, Human Resource Machine faz fisicamente o jogador ligar uma seta entre o início do loop e o final do mesmo. Contudo, enquanto Human Resource Machine aborda laços através de instruções de pulo (tal como em linguagens Assembly), o nosso jogo permite que blocos de instruções sejam definidos e que esses blocos sejam executados repetidamente por um número fixo de vezes, tal como um laço definido por um *for*. Desta forma, não exigimos que o jogador aprenda o conceito de laço como sendo aplicação de pulo entre instruções.

⁵ **Human Resource Machine.** Steam, 2015. Disponível em: https://store.steampowered.com/app/375820/Human_Resource_Machine/. Acesso em: 13 dez. 2020.

Figura 3: Exemplo de fase do jogo “Human Resource Machine”



Fonte: Página da Steam, acessada em 2020

Outros jogos foram consultados como forma de referência, como é o caso de TIS-100⁶ e Opus Magnum⁷. Respectivamente, estes jogos se tratam de programar instruções em uma linguagem de programação parecida com Assembly e programar mecanismos para montar cadeias carbônicas de elementos através de uma interface gráfica de arrastar instruções para uma trilha conforme se inserem partes programáveis na máquina. Os dois jogos se utilizam de contextos diferentes de apresentar a ideia de montar algoritmos e duas formas distintas de contar uma história de modo a gerar algum engajamento por parte do jogador além da parte mecânica, sendo TIS-100 uma história de mistério sobre um computador misterioso e Opus Magnum um drama sobre um alquimista à serviço da corte. Estes jogos nos inspiram pelo modo como eles dosam informações ao jogador, escalando lentamente a dificuldade entre problemas para acostumar o jogador com formas de pensar distintas entre as fases, para que então eventualmente os jogadores sejam compelidos a atacar problemas mais complicados. Opus Magnum também mostrou, involuntariamente, o potencial por trás de comunidades; fãs do jogo que gostam tanto e se reúnem em ambientes virtuais para mostrarem as próprias soluções para fases do jogo, competindo entre si para ver quem monta os mecanismos mais

⁶ TIS-100. Steam, 2015. Disponível em: <https://store.steampowered.com/app/370360/TIS100/>. Acesso em: 13 dez. 2020.

⁷Opus Magnum. Steam, 2017. Disponível em: https://store.steampowered.com/app/558990/Opus_Magnum/. Acesso em: 13 dez. 2020.

eficientes ou estéticos, desta forma provando mais uma vez o poder de gerar engajamento no jogador.

3 PROJETO DE JOGO EDUCATIVO PARA AUXÍLIO NO ENSINO DE PENSAMENTO COMPUTACIONAL

Neste capítulo, detalharemos o projeto do jogo desenvolvido. Na seção 3.1, apresentaremos os métodos utilizados para a concepção da proposta do jogo educativo. Na seção 3.2, entraremos em maiores detalhes sobre o objetivo pedagógico do nosso trabalho e em como esses objetivos são contemplados através do jogo. Na seção 3.3, analisaremos os elementos lúdicos do jogo através do modelo MDA. Por fim, na seção 3.4, mostraremos como que o jogo se organiza para passar a sua mensagem de modo a fazer sentido dentro da proposta didática.

3.1 CONCEITO

Nesta seção, apresentaremos a ferramenta utilizada para solidificarmos a proposta de jogo educativo que compõe este trabalho. Na subseção 3.1.1, explicaremos sobre o canvas utilizado especificamente para o planejamento de um jogo educativo endógeno. Na subseção 3.1.2, brevemente discorreremos sobre o documento de *game design* criado a partir do canvas utilizado no planejamento do jogo.

3.1.1 Game Design Canvas para jogos educativos endógenos

Um canvas é uma ferramenta que permite reunir as informações principais sobre algo a ser desenvolvido em um único painel, de modo a tornar informações importantes sempre à vista e agrupadas graficamente de uma forma que faça sentido. Exercendo um importante papel no campo de engenharia de software, canvas são ferramentas que permitem organizar times de variados tamanhos com uma miríade de propósitos.

De modo a auxiliar times focados em desenvolvimento de jogos educativos, Taucei (2019) propôs o Endo-GDC, um modelo de canvas voltado para a “concepção de jogos educativos endógenos”. O Endo-GDC é uma ferramenta mais apropriada para o nosso trabalho do que outros canvas mais estabelecidos para desenvolvimento de jogos uma vez que

este canvas exige não apenas que exista um objetivo pedagógico por trás do projeto, mas que o jogo integre o conteúdo de aprendizado ao seu próprio funcionamento.

O Endo-GDC concebido e utilizado para a criação do jogo pode ser encontrado no Apêndice A.

3.1.2 Game Design Document

Game Design Document (GDD) é um documento importante para o desenvolvimento de um jogo, seja ele educativo ou não. É nesse documento onde são listados todos os detalhes relevantes para o desenvolvimento de um jogo, desde especificações técnicas até a filosofia por trás do desenvolvimento. Um GDD é organizado de modo a fazer sentido para a equipe em questão, tendo um formato que varia de acordo com o time de desenvolvedores e com o projeto.

Uma vez que nós trabalhamos com o Endo-GDC (TAUCEI, 2019) para o planejamento do jogo, o passo seguinte foi escrever o GDD organizado de modo a refletir os campos do Endo-GDC. Cada capítulo do GDD reflete ao que o Endo-GDC chama de “bloco”, e cada capítulo é dividido por seções correspondentes à divisão do Endo-GDC. O GDD resultante dos nossos esforços utilizado na concepção do jogo pode ser encontrado no Apêndice B.

No GDD, ficaram definidas também as especificações técnicas do jogo, como por exemplo as ferramentas que seriam utilizadas em sua confecção e as plataformas-alvo. No caso deste trabalho, o jogo foi feito em maior parte na Unity Engine⁸, com a arte sendo feita no programa pago Aseprite⁹, com foco em celulares Android e computadores pessoais.

3.2 OBJETIVO PEDAGÓGICO

Nesta seção, entraremos em detalhes sobre os objetivos que o nosso projeto tem a nível pedagógico. Por mais que o trabalho seja um jogo eletrônico, ele ainda possui uma proposta educativa ao tentar auxiliar no ensino de PC. Na subseção 3.2.1, listaremos os

⁸ **Unity Engine.** Unity Technologies, 2021. Disponível em: <https://unity.com>. Acesso em: 05 mar. 2021.

⁹ **Aseprite.** Steam, 2016. Disponível em: <https://store.steampowered.com/app/431730/Aseprite/>. Acesso em: 05 mar. 2021

conceitos que serão diretamente ensinados ao jogador no ato de jogar o jogo. Na subseção 3.2.2, explicaremos quais são as habilidades que almejamos passar ao jogador através do conteúdo pedagógico. Por fim, na subseção 3.2.3, faremos a relação entre o conteúdo pedagógico passado através do jogo em si e os objetivos pedagógicos que temos para o nosso projeto.

3.2.1 Conteúdo Pedagógico

Os conteúdos pedagógicos apresentados estão alinhados a currículos reais de cursos de programação para o Ensino Médio, tendo como maior referência a ementa do curso eletivo de C++ proposto pelo programa K12 (K12.COM, acessado em 2020) para turmas de ensino médio. A ementa descreve a ordem na qual os conteúdos serão passados, que foi levada em consideração na hora de distribuir os conteúdos ao longo do jogo. Contudo, a ementa em questão não aborda concorrência, além de abordar conceitos que o jogo não cobre de forma direta, como strings e funções. Desta forma, o jogo seguirá em grande parte uma ementa conhecida e utilizada em salas de aula, mas sem se comprometer a passar todos os conteúdos da mesma ementa.

Ao longo das fases do jogo, os conteúdos pedagógicos a serem transmitidos serão:

- Conceitos básicos de algoritmo: O jogo ensinará ao jogador a construir uma “receita” que será seguida fielmente pelos personagens. Para tal, o jogador inserirá instruções básicas em uma trilha de instruções, que servirá de artifício visual para entender a ideia de instruções a serem executadas ao longo do tempo. Cada trilha existente terá um personagem correspondente na tela. A ementa supracitada não possui uma seção dedicada sobre pensar em algoritmos, sendo essa ideia exercitada ao longo do curso através do ensino de matérias pontuais;
- Variáveis: O jogo não permitirá que o jogador declare variáveis de modo a diminuir a complexidade de elementos na tela, mas abordará o conceito através dos objetos e animais segurados pelos personagens. Cada personagem pode estar ou não segurando algo em sua boca, colocado lá através de uma instrução utilizada pelo jogador;
- Condicionais: O jogo não permite a declaração de condicionais do tipo *if-then-else* de forma livre, uma vez que isso adicionaria um grande nível de complexidade visual.

Contudo, o jogador tem acesso a instruções que se comportam diferente de acordo com as variáveis mencionadas acima, introduzindo-o ao conceito de condicionais;

- Laços: O jogo ensinará como definir laços de instruções que são repetidos por um número pré-determinado de vezes, fazendo o jogador visualmente definir em que momento da trilha de instruções o laço acontece. O ensino de laços na ementa do curso de C++ para K12 é dado através de duas seções diferentes, ensinando primeiro na forma de `while()` e depois na forma de um `for()`. No jogo, será abordada apenas a forma de `for()`, visto que garantiria ao jogador mais liberdade na declaração da comparação sem grande complexidade visual;
- Concorrência: O jogo demonstrará, de forma prática, o que é e como usar concorrência para a resolução de problemas. Em determinadas fases, o jogador terá acesso a dois personagens que executarão as instruções inseridas em cada trilha de forma simultânea, e terá alguma liberdade de como usar os dois personagens na resolução da tarefa;
- Barreira: O jogo fará o jogador controlar fluxos de instruções concorrentes através de barreiras, definidas através de uma instrução especial.

3.2.2 Objetivos de Aprendizado

A partir dos conteúdos explicitamente abordados na seção 3.2.1, o jogo terá como objetivo transmitir aprendizado de dois dos quatro pilares (BBC.CO.UK/BYTESIZE) que compõem o pensamento computacional. Isto implica que o jogo **não** exercitará os pilares denominados abstração e reconhecimento de padrões, que consistem, respectivamente, na interpretação de informações reais de modo a poder criar soluções computacionais usando informações úteis para a compreensão dos problemas para os quais as soluções serão feitas e na capacidade de encontrar semelhanças entre problemas de modo a usar algoritmos já conhecidos.

Os pilares de PC a serem aprendidos com o jogo são dois: a decomposição de problemas e a concepção de algoritmos. No que tange a decomposição de problemas, este pilar consiste em ser capaz de decompor o problema de uma fase em problemas menores

capazes de serem solucionados pelos recursos dados ao jogador. Portanto, consideraremos prova de que o jogador domina suficientemente esse pilar se este for capaz de, dentro de um problema complexo, identificar problemas menores que compõem o problema maior.

Por sua vez, a concepção de algoritmos é o pilar principal do jogo, e deve ser ensinado conforme o jogador conclui as fases. Conceber um algoritmo consiste em, dentro de um número limitado de passos, sequenciar instruções para os personagens jogáveis dentro da fase, declarar laços de instruções para serem repetidos um número fixo de vezes e executar o algoritmo para *feedback* posterior. Consideraremos prova de que o jogador domina esse pilar o suficiente uma vez que este seja capaz de interpretar o que é pedido em uma fase, analisar as limitações impostas pela fase e conceber um algoritmo utilizando as instruções fornecidas pela fase.

A decisão de limitar o número de conteúdos pedagógicos a apenas uma parte do pensamento computacional se dá pela própria natureza de cada pilar. Cada um dos pilares de PC são exercitados de forma diferente em ambientes didáticos (BARR e STEPEHSON, 2011), o que nos fez optar por um jogo que abordasse um número limitado de pilares sem se tornar desnecessariamente convoluto.

O Endo-GDC recomenda que, ao estabelecer objetivos de aprendizado para o seu jogo, que se utilize a Taxonomia Revisada de Bloom (KRATHWOHL, 2002) para descrever os objetivos de forma simples e objetiva. A Taxonomia de Bloom se trata de um modelo para classificação de objetivos de aprendizado, de modo a facilitar a comunicação entre educadores de diferentes campos. (KRATHWOHL, 2002) A escolha da versão revisada da taxonomia para aplicação no trabalho se dá pela maior clareza que a revisão provê ao usuário na hora de se estabelecer objetivos, tratando o conteúdo a ser aprendido como substantivos e os processos para se adquirir o aprendizado enquanto verbos. (TAUCEI, 2019)

Como forma de manter a definição dos objetivos de aprendizado objetiva para este projeto, utilizamos os verbos sugeridos pela Taxonomia Revisada de Bloom para definir os objetivos almejados. A definição desses verbos emergiu dos dois pilares mencionados, que são os nossos objetivos finais, fazendo assim da definição dos verbos mais uma formalidade de modo a manter as definições claras e acessíveis.

Portanto, os objetivos de aprendizado listados são quatro: (i) reconhecer subproblemas, (ii) interpretar o objetivo da fase, (iii) analisar as limitações da fase, e (iv) conceber algoritmos.

3.2.3 Relação entre objetivos de aprendizado e conteúdo pedagógico

Os pilares a serem aprendidos pelo jogador possuem relação direta com os conteúdos pedagógicos abordados, tendo em vista o alinhamento entre os objetivos do jogo e a intenção de transmitir aprendizado sobre pensamento computacional. A relação explícita entre conteúdos e os objetivos de aprendizado podem ser vistos no quadro 2.

Quadro 2: Relação entre conteúdos pedagógicos e objetivos de aprendizado

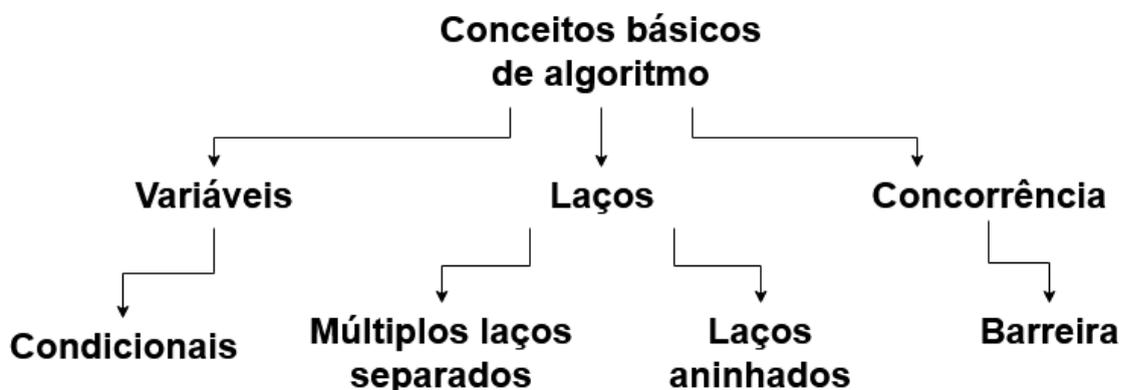
	reconhecer subproblemas	interpretar o objetivo da fase	analisar as limitações da fase	conceber algoritmos.
Conceitos básicos de algoritmo		É necessário ter a compreensão do que é pedido em uma fase para construir um algoritmo para resolver o problema de uma fase.	Para construir um algoritmo para resolver o problema de uma fase, é necessário analisar os objetos dentro de uma fase e como eles podem contribuir para uma solução	A compreensão de conceitos básicos de algoritmo é exercitada através da construção de algoritmos progressivamente mais complexos conforme o jogador passa de fase
Variáveis			É importante que o jogador reconheça onde utilizar cada variável em uma determinada fase	Compreender o modo como variáveis são interpretadas pelo jogo é importante para ter domínio sobre os elementos do jogo e montar algoritmos úteis para concluir as fases
Condicionais			É importante que, ao analisar os objetos da fase, o jogador compreenda o comportamento que as condicionais terão ao utilizá-las nos objetos	Compreender o jeito como condicionais são utilizadas no jogo é importante para se utilizar uma das instruções que o jogador usa na hora de conceber algoritmos
Laços	Problemas que podem ser divididos em múltiplos subproblemas iguais podem ser resolvidos com o uso de laços		Certas fases, por causa da limitação de instruções em uma trilha, exigem que o jogador compreenda o que é pedido de modo a declarar laços que resolvam a fase dentro das limitações impostas	Compreender laços é importante para que se monte algoritmos mais complexos, visto que a limitação de instruções no jogo pode ser superada através de laços

Concorrência	Alguns problemas são compostos de dois ou mais subproblemas que podem ser resolvidos de forma independente, possibilitando soluções com uso de concorrência	Compreender concorrência é importante para que se possa realizar objetivos que dependem de cada um dos personagens	Compreender concorrência é importante para que se possa resolver determinadas fases que necessitam que se use mais de um personagem ao mesmo tempo, devido à limitação de instruções por personagem	Compreender concorrência é importante para que se monte algoritmos que façam bom uso da existência de mais de um personagem na fase, de modo que os dois consigam minimizar a quantidade de instruções executadas por cada um dos personagens
Barreira	Alguns subproblemas devem ser sincronizados entre os personagens, tornando necessário saber como usar barreiras para tal		Algumas fases levam os personagens a uma condição de corrida, que devem ser mitigadas dentro do jogo com o uso de barreiras.	Barreiras são importantes na hora de controlar o fluxo de execução de algoritmos que rodam de forma concorrente entre si, de modo a impedir que os algoritmos se atrapalhem e induzam um ao outro a uma solução errada para o problema em questão

Fonte: Acervo do autor (2020)

Na Figura 5, especulamos sobre a precedência entre os conteúdos pedagógicos reconhecidos na forma de uma árvore. Nesse diagrama, interpretamos que um conteúdo só faz sentido ser ensinado se e somente se o conteúdo diretamente acima tiver sido ensinado previamente, uma vez que um conhecimento serve de base para o outro.

Figura 5: Precedência entre conteúdos pedagógicos



Fonte: Acervo do autor (2020)

3.3 O JOGO SEGUNDO O MODELO MDA

Nesta seção, analisaremos os elementos lúdicos que compõem o jogo descrito neste trabalho final. Para realizar tal tarefa, faremos uso do modelo MDA proposto por Hunicke, LeBlanc e Zubek em 2001 para descrever o funcionamento do jogo assim como descrito no canvas mencionado na seção 3.1. Na subseção 3.3.1, falaremos sobre as mecânicas do jogo, isto é, as regras pelas quais o jogador interage com o jogo. Na subseção 3.3.2, falaremos sobre as dinâmicas do jogo, as formas como o jogo se comportará diante da interação do jogador durante a sua execução. Por fim, na subseção 3.3.3, falaremos sobre a estética do jogo, sobre a resposta emocional que o jogo visa causar ao jogador.

3.3.1 Mecânicas

As mecânicas dentro do modelo MDA são os elementos fundamentais que compõem o jogo. Esses elementos são todas as regras, comportamentos e ações que um jogador pode tomar dentro do contexto do jogo.

No nosso jogo, as mecânicas listadas no Endo-GDC foram: trilha de instruções de tamanho limitado, definir laços entre instruções que serão executadas por um número fixo de vezes, personagens jogáveis diferentes, árvores, objetos empurráveis, movimentar personagens pelo mapa, pegar objetos e barreira. O Endo-GDC não só pede a especificação de mecânicas de forma objetiva, como também que exista uma relação direta entre as mecânicas do jogo e os objetivos de aprendizado de forma a garantir que o jogo seja endógeno. A relação está descrita no quadro 3.

Quadro 3: Relação entre mecânicas do jogo e objetivos de aprendizado

	reconhecer subproblemas	interpretar o objetivo da fase	analisar as limitações da fase	conceber algoritmos.
Trilha de instruções de tamanho limitado				A trilha de instruções é onde o jogador monta os algoritmos, o que viabiliza a montagem de soluções algorítmicas ao mesmo tempo em que estabelece a

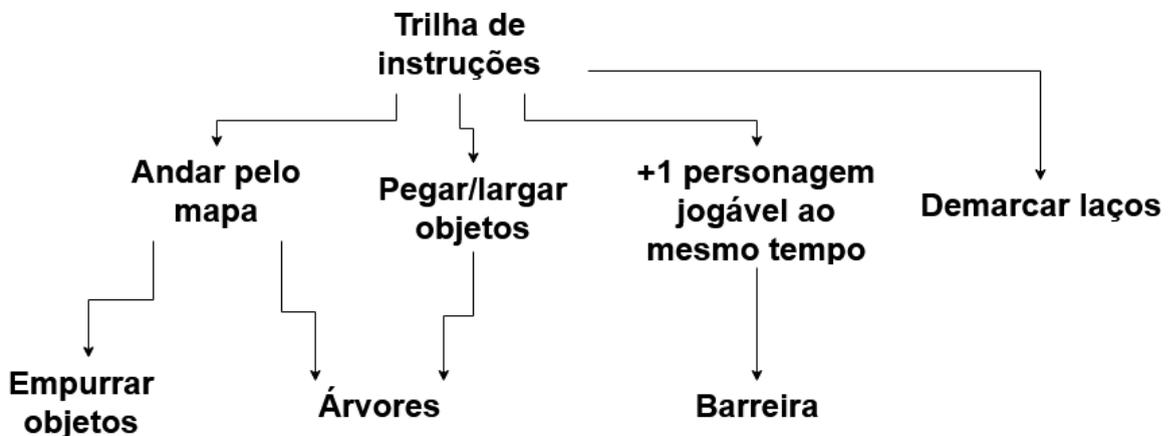
				limitação de instruções diferentes em uma dada solução
Andar pelo mapa		Os objetivos de algumas fases envolvem fazer os personagens andarem até lugares específicos	É importante analisar limitações da fase, como paredes ou objetos não empurráveis no meio do mapa, que podem dificultar o uso da instrução de andar	Andar pelo mapa é uma das instruções que pode ser colocada na trilha para montar o algoritmo que resolve a fase
Empurrar objetos			A mecânica de empurrar alguns objetos com um dos personagens jogáveis realça a diferença entre personagens e obriga o jogador a levar em consideração essa diferença na hora de programar as soluções algorítmicas de cada um	Determinadas fases requerem algoritmos que tirem objetos do caminho com esta mecânica para serem resolvidas.
Pegar/largar objetos		Os objetivos de algumas fases envolvem largar objetos em lugares específicos	É importante analisar a presença de objetos pegáveis na fase e os lugares apropriados onde se deve largar esses mesmos objetos	Pegar/largar objetos é uma das instruções que pode ser colocada na trilha para montar o algoritmo que resolve a fase
Árvores			É preciso compreender como os diferentes personagens interagem com árvores para resolver fases que possuam uma ou mais árvores	Determinadas fases requerem algoritmos que incluam fazer o personagem subir em árvores para cumprir o objetivo
Laços de instruções	Problemas que podem ser divididos em múltiplos subproblemas iguais podem ser resolvidos com o uso de laços		Certas fases, por causa da limitação de instruções em uma trilha, exigem que o jogador compreenda o que é pedido de modo a declarar laços que resolvam a fase dentro das limitações impostas	Compreender laços é importante para que se monte algoritmos mais complexos, visto que a limitação de instruções no jogo pode ser superada através de laços
Dois personagens jogáveis	Algumas fases são compostas de dois ou mais subproblemas que podem ser resolvidos de forma independente, o que, junto da limi-	Os objetivos de algumas fases envolvem usar necessariamente dois personagens jogáveis	Usar dois personagens ao mesmo tempo é importante para que se possa resolver determinadas fases, devido à limitação de instruções por personagem ou diferenças entre personagens jogáveis	Determinadas fases requerem que se faça bom uso da existência de mais de um personagem na fase, de modo que os dois consigam realizar objetivos que um personagem sozinho não consegu-

	tação da trilha de instruções, exige o uso dos dois personagens jogáveis agindo de forma concorrente			ria
Barreiras	Alguns subproblemas devem ser sincronizados entre os personagens, tornando necessário saber como usar barreiras para tal		Algumas fases levam os personagens a uma condição de corrida, que devem ser mitigadas dentro do jogo com o uso de barreiras.	Barreiras são importantes na hora de controlar o fluxo de execução de algoritmos que rodam de forma concorrente entre si, de modo a impedir que os algoritmos se atrapalhem e induzam um ao outro a uma solução errada para o problema em questão

Fonte: Acervo do autor (2020)

As mecânicas são introduzidas ao jogador de forma progressiva, aumentando a complexidade das fases apresentadas uma vez que o jogador possui a cada mecânica aprendida mais ações possíveis de serem tomadas. Mas também mecânicas por vezes precisam de outras mecânicas para fazerem sentido. Um exemplo disso é a mecânica de empurrar, que não faz sentido ser introduzida ao jogador antes que o mesmo aprenda a fazer um personagem andar pela tela. De modo a manter a sobriedade na ordem em que mecânicas são introduzidas, a figura 6 descreve as relações de dependência entre mecânicas distintas.

Figura 6: Precedência entre mecânicas do jogo



Fonte: Acervo do autor (2020)

3.3.2 Dinâmicas

Dinâmicas no modelo MDA são o resultado do funcionamento das ações que o jogador toma através das mecânicas. Diferente das mecânicas, as dinâmicas só existem uma vez que o jogador joga o jogo, tendo em vista que estas emergem da interação do jogador com as mecânicas (SALEN; TEKIMBAS; ZIMMERMAN, 2006). Por serem produto direto das mecânicas, as dinâmicas listadas nesta seção possuem também relação com os objetivos pedagógicos estipulados na seção 3.2, embora a relação não seja direta.

No nosso Endo-GDC, as dinâmicas descritas foram: administrar dois personagens de forma concorrente, subir em árvores, carregar objetos ou outros animais e montar algoritmos. Todas estas dinâmicas, contudo, podem ser resumidas em um dos itens da tabela proposta por Jukka Vahlo (et al., 2017) onde são listadas as trinta e três dinâmicas comuns em jogos eletrônicos. O item em questão é o primeiro da tabela, a “resolução de problemas que requerem lógica, raciocínio ou pensamento analítico”.

3.3.3 Estética

A estética no modelo MDA é a resposta emocional causada ao jogador através da interação com o jogo. Como a estética é o primeiro contato de um jogador com o jogo, é importante que ela seja definida em termos não abstratos para que se possa trabalhar de modo a construir as experiências pretendidas ao jogador. A estética de um jogo emerge das dinâmicas; ou seja, as emoções que o jogador sentirá ao jogar o jogo virão diretamente de como o jogo vai se comportar ao ser jogado (SALEN; TEKIMBAS; ZIMMERMAN, 2006).

A estética dentro do modelo MDA, descrita primeiramente por Hunicke, LeBlanc, Zubek (2001, p.2), é composta por uma taxonomia limitada que admitidamente não engloba todas as possibilidades estéticas de um jogo. No nosso jogo, as estéticas listadas no Endo-GDC foram: meiguice, calma para resolver as fases, satisfação após resolver uma fase. Dentro da taxonomia proposta por Hunicke, LeBlanc, Zubek (2001, p.2), a estética que nós buscamos seria encontrada nas palavras “Desafio”, “Sensação” e “Submissão”.

A taxonomia de Hunicke foi explorada mais a fundo por Abbott (2010), que entrou em detalhes sobre o significado de cada termo encontrado no grupo de palavras. “Desafio” foi definida como a sensação de dominar um desafio de modo a resolvê-lo de forma proficiente. “Sensação”, por sua vez, descreve estímulos sensoriais prazerosos através de *feedback* (tanto visual quanto auditivo) pelas ações tomadas. Por fim, “Submissão” é definida como um jogador voluntariamente se submetendo a um cenário limitado de modo a ter a experiência do jogo; no nosso caso, isso seria o lidar com um conjunto limitado de instruções para a resolução de cada fase.

3.4 ESTRUTURA DO JOGO

Nesta seção, analisaremos como o jogo foi estruturado de modo a passar o conteúdo pedagógico. Na seção 3.4.1, revisamos como funciona uma fase comum dentro do jogo, do momento em que o jogador escolhe jogar uma fase até o momento em que ele completa o objetivo imposto. Por fim, na seção 3.4.2, discorreremos sobre um possível plano de fases, levando em consideração a precedência entre conteúdos pedagógicos e justificando o conteúdo de cada seção.

3.4.1 Estrutura de uma fase dentro do jogo

As fases do jogo funcionam todas praticamente da mesma forma, de modo a manter a consistência na interação do jogador com as mecânicas do jogo e não exigir que o jogador reaprenda a jogar a cada fase. Caso a fase seja a primeira de algum conjunto de fases agrupada por introduzir uma mecânica ou conteúdo pedagógico diferente e o jogador estiver jogando a fase pela primeira vez, um vídeo curto na forma de tutorial demonstra através de um exemplo a mecânica nova.

Independente de qual for a fase dentro do seu grupo de fases, todas as fases se dão como descrito a seguir. O jogador tem acesso à visão inteira do mapa da fase, a uma descrição do objetivo e todo o tempo do mundo para conceber uma resposta. Contudo, desde o início da fase, o jogador é livre para arrastar instruções, representadas por artefatos visuais, até a trilha de instruções localizada no canto da tela.

Figura 7: Tela inicial da primeira fase do primeiro grupo de fases



Fonte: Acervo do autor (2020)

Assim que o jogador se der por satisfeito com a solução concebida, o jogador deve apertar no botão “pronto” presente na figura 7 para dar continuidade. Se a mecânica de determinar laços de instruções não tiver sido ensinada, o jogador será levado diretamente para a execução do algoritmo. Caso contrário, o jogador terá a liberdade de definir laços de instruções com um número finito de repetições. Para sinalizar a mudança nas possibilidades de interação, a trilha de instruções de cada personagem presente na fase apresentará símbolos os quais arrastar para formar laços entre as instruções pré-existentes, como visto na figura 8. Nesta interface também o jogador tem toda a liberdade de retornar para a tela anterior e fazer algum ajuste às instruções presentes na trilha e voltar a ela.

Figura 8: Tela da interface de laços da primeira fase do quinto grupo de fases, perguntando por quantas vezes um laço de instruções específico deve se repetir



Fonte: Acervo do autor (2020)

Uma vez que o algoritmo for executado, o jogo rodará por todas as instruções até que um dos dois cenários ocorram: ou o personagem cumpre o objetivo proposto pela fase seguindo o algoritmo concebido pelo jogador, ou o jogador fornecerá ao personagem um algoritmo incapaz de resolver a fase. Em caso de sucesso, o jogador terá a opção de voltar ao menu principal ou jogar a mesma fase novamente, como visto na figura 9. Então o jogador poderá voltar ao menu principal e selecionar a próxima fase para jogar.

Figura 9: Tela de vitória da primeira fase do quinto grupo de fases



Fonte: Acervo do autor (2020)

Caso o algoritmo do jogador tenha sido insuficiente, uma mensagem de erro aparecerá na tela, explicando através de uma mensagem simples qual foi o erro. O erro pode ter levado algum personagem a bater contra uma das extremidades do mapa, bater contra um objeto não-empurrável, bater no outro personagem ou não cumprir o objetivo ao final da execução da fase. Em qualquer um dos casos, o jogador é dado a oportunidade de rejogar a fase, cenário no qual a trilha de instruções com o algoritmo errado se manterá na tela para que o jogador possa trabalhar em cima da solução incorreta, ou voltar ao menu principal.

Figura 10: Tela de derrota da primeira fase do quinto grupo de fases, explicitando o motivo da derrota



Fonte: Acervo do autor (2020)

3.4.2 Planejamento de fases

O jogo é dividido em fases fechadas, onde cada fase terá um problema a ser resolvido pelo jogador. As fases servem a um de dois possíveis propósitos: ensinar o jogador alguma mecânica ou dinâmica do jogo, ou testar os conceitos ensinados pelas fases anteriores.

As fases estão organizadas em grupos, reunindo fases que ensinam e exercitam as mesmas mecânicas ou os mesmos conteúdos pedagógicos. Desta forma, o ensino tanto dos conhecimentos necessários para se jogar quanto os conteúdos pedagógicos que se quer ensinar são passados de forma comedida, com cuidado para não sobrecarregar o jogador em dado momento.

A precedência entre conteúdos pedagógicos foi fortemente baseada em currículos escolares usados como base. Em específico, foram utilizados os currículos do curso de C++ oferecido pelo K12 e o de computação previsto pelo currículo nacional da Inglaterra como base para organizarmos a ordem de apresentação dos conteúdos pedagógicos.

De modo a tornar o jogador confortável tanto com as mecânicas do jogo enquanto atividade lúdica quanto com o conteúdo a ser aprendido, as fases do jogo foram divididas em seis seções. Cada seção tem um número variado de fases, variando de duas até seis fases, e um tópico específico. Ao final, nós planejamos vinte e três fases para a versão final do jogo.

A primeira seção do jogo é composta por quatro fases, onde o jogador aprende o conteúdo mais básico e o mínimo de interação possível de modo a poder jogar o jogo. Até o final desta seção, o jogador aprendeu a alinhar instruções para montar algoritmos simples e terá sido introduzido à ideia de fazer isso de forma concorrente.

A segunda seção do jogo é composta por duas fases, e é onde o jogador aprende sobre a existência de caixas empurráveis no jogo. Esta seção não introduz nenhum conteúdo pedagógico novo, apenas reforça conceitos básicos de algoritmos enquanto introduz uma nova mecânica. Dentre os dois personagens existentes, apenas a cachorra é capaz de empurrar caixas, enquanto elas servem de obstáculo para o gato. Isto introduz uma diferença entre personagens, justificando a existência de dois personagens distintos através de mecânica. O papel desta seção é ensinar uma mecânica do jogo enquanto reforça a ideia de pensar em algoritmos, de modo a solidificar os conceitos básicos de algoritmos.

A terceira seção do jogo é composta por quatro fases, e é onde o jogador aprende a usar a instrução de pegar e largar objetos. Esta seção introduz as ideias de variável e de condicional, mesmo que de forma indireta. A instrução de pegar ou largar objetos age de acordo com a condição em que o personagem que a executa se encontra; se o personagem está segurando um objeto ou animal, usar a função o fará largar o que estiver segurando. Caso contrário, se a instrução for executada na direção de algo que pode ser pego, esse objeto ou animal será pego pelo personagem em questão. Não só isso, mas pelo menos uma fase pede para que o jogador atente ao tipo do que se quer pegar, fazendo diferença entre pegar um objeto ou um animal. O papel desta seção é ensinar uma mecânica do jogo que permitirá mais variedade nos algoritmos montados enquanto introduz a noção de variável com diferentes tipos e instruções condicionais.

A quarta seção do jogo é composta por três fases, e é onde o jogador aprende a lidar com árvores dentro do jogo. Esta seção não introduz nenhum conteúdo pedagógico novo, apenas reforça conceitos básicos enquanto introduz a mecânica de subir em árvores. Dentre os dois personagens existentes, apenas o gato é capaz de subir em árvores, enquanto elas servem de obstáculo para a cachorra. Isto introduz uma diferença entre personagens, reforçando que os dois personagens possuem habilidades distintas e são úteis cada um à sua maneira. O papel desta seção é ensinar uma mecânica do jogo enquanto reforça a ideia de pensar em algoritmos, de modo a solidificar os conceitos básicos de algoritmos.

A quinta seção do jogo é composta por seis fases, e é onde o jogador aprende a definir laços entre instruções. Esta seção introduz não só o conteúdo pedagógico, mas também a mecânica de definir laços de forma gráfica arrastando as setas na trilha de instruções, expandindo assim o número de instruções que podem ser executadas em uma fase. O papel desta seção é ensinar uma nova mecânica enquanto introduz o conceito de laço para o jogador, dado que este já está acostumado com a noção de montar algoritmos com base em instruções atômicas.

A sexta e última seção do jogo é composta por apenas três fases, e é onde o jogador aprende sobre barreiras de instruções. Esta seção introduz a instrução de barreira; uma instrução onde um dos personagens interrompe o próprio fluxo de execução e precisa que o outro personagem execute a mesma função para continuar o próprio fluxo de execução. Desta forma, o jogador aprenderia que concorrência não tem só vantagens, mas também exige às vezes um nível de cuidado na hora de lidar com mais de um personagem realizando tarefas ao

mesmo tempo. O papel desta seção é ensinar uma nova mecânica enquanto introduz o conceito de barreira para o jogador, uma vez que este teve o jogo inteiro para se acostumar com a noção de concorrência.

O quadro 4 resume a relação entre as seções de fases planejadas para a versão final do jogo e os conteúdos pedagógicos de cada seção.

Quadro 4: Relação entre seção do jogo e conteúdo pedagógico

SEÇÃO DO JOGO	NÚMERO DE FASES	CONTEÚDOS PEDAGÓGICOS EXERCITADOS
Primeira	4	Conceitos básicos de algoritmos; paralelismo
Segunda	2	Conceitos básicos de algoritmos; paralelismo
Terceira	4	Conceitos básicos de algoritmos; paralelismo; condicionais; variáveis
Quarta	3	Conceitos básicos de algoritmos; paralelismo; condicionais; variáveis
Quinta	6	Conceitos básicos de algoritmos; paralelismo; condicionais; variáveis; laços
Sexta	3	Conceitos básicos de algoritmos; paralelismo; condicionais; variáveis; laços; barreiras

Fonte: Acervo do autor (2020)

4 VALIDAÇÃO DO PROJETO

Neste capítulo, falaremos sobre o método utilizado para a validação do jogo enquanto uma ferramenta pedagógica apropriada para ensinar noções de PC a crianças e adolescentes. Na seção 4.1, descreveremos as condições sob as quais a validação aconteceu e quais concessões precisaram ser feitas. Na seção 4.2, nós entramos em maiores detalhes sobre o perfil dos voluntários consultados no processo de validação. Na seção 4.3, nós entramos em maiores detalhes sobre a pesquisa feita, explicando o objetivo de cada pergunta feita e o processo pelo qual as respostas foram tratadas. Na seção 4.4, apresentaremos os resultados obtidos através da nossa pesquisa com voluntários. Por fim, na seção 4.5, analisaremos as críticas e sugestões recebidas acerca do jogo pelos voluntários.

4.1 DESCRIÇÃO DO CONTEXTO DA VALIDAÇÃO

A validação do trabalho se deu no meio de uma pandemia global que afetou o funcionamento de escolas e a nossa capacidade de diretamente irmos atrás dos voluntários. Além disso, como o nosso público alvo é composto por menores de idade, existem questões éticas que tomariam tempo demais caso buscássemos voluntários dentro da faixa etária do nosso público pretendido.

De modo a compensar essa questão, nós buscamos professores de diversas matérias com experiência dando aula para diversas faixas etárias de modo a coletarmos *feedback* relevante sobre o potencial pedagógico do jogo. A ideia vem da necessidade de confirmar se o jogo pode servir, aos olhos de um professor, como uma forma de ajudar no ensino de um conteúdo, por mais que o mesmo não seja um substituto para uma aula formal. Além disso, professores são maiores de idade e capazes de consentir com participação voluntária em pesquisas sem maiores complicações.

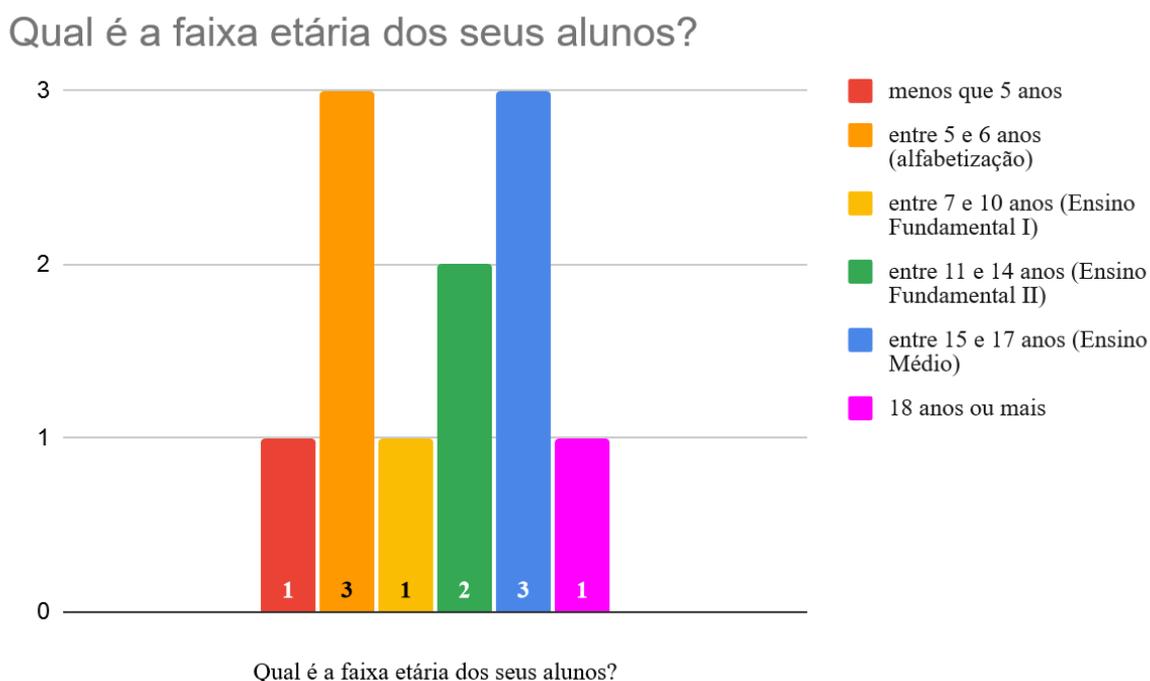
A validação foi conduzida do dia 14 de Setembro de 2020 ao dia 3 de Novembro de 2020, e contou com a participação de onze professores.

4.2 PERFIL DOS VOLUNTÁRIOS

A priori, apenas professores de Ensino Fundamental 1 e Ensino Fundamental 2 de matemática e computação seriam abordados acerca do projeto, por terem experiência com a faixa etária proposta e com as áreas de conhecimento que costumam abordar pensamento computacional. Contudo, de modo a concluir a pesquisa em um espaço de tempo razoável com um número suficiente de respostas em um momento de pandemia, professores de outras matérias e etapas da educação também foram contemplados, de modo a coletar o maior número de perspectivas acerca da utilidade prática do jogo possível.

Ao final do período de dois meses, onze voluntários contribuíram para a pesquisa. Destes onze voluntários, apenas três estão dentro da faixa etária estabelecida de sete a onze anos. Fora dessa faixa etária, três professores são professores de alfabetização, outros três professores são professores de Ensino Médio, uma professora dá aula para alunos de pré-escola e, por fim, um professor dá aula para o Ensino Superior. A relação de professores por faixa etária dos seus alunos pode ser vista no Gráfico 1.

Gráfico 1: Relação de professores por faixa etária dos alunos



Fonte: Acervo do autor (2020)

Os professores que lecionam para crianças muito pequenas (com menos de cinco anos) ou adultos (dezoito anos ou mais) foram os últimos professores a serem contactados. Contudo, até recebermos as respostas, acreditávamos serem professores de faixas etárias mais próximas do nosso público-alvo. De qualquer forma, esses professores darem aula para alunos distantes da faixa etária pretendida acabou nos servindo na hora de validar o trabalho; entraremos em maiores detalhes na seção 4.4.

Neste grupo de professores, muitos deles lecionam mais de uma matéria durante o ano letivo. A maioria, composta por sete dos onze professores, leciona matemática dentro de sala de aula de alguma forma. As segundas matérias mais representadas, cada uma com quatro professores dentro do conjunto, são ciências¹⁰ e português. As terceiras matérias mais representadas, com três professores cada, são história e geografia. Por fim, com um professor de cada, um professor dedicado de pensamento computacional e um professor de termodinâmica clássica responderam ao questionário.

Dentre esses onze professores, seis professores alegaram usar jogos educativos em sala de aula no momento da pesquisa. Três outros professores expressaram o desejo de usar jogos educativos em sala de aula, mas nunca o fizeram alegando falta de recursos nas instituições de ensino, falta de tempo no calendário escolar e a complicação de organizar uma atividade lúdica em uma turma grande. Dos outros dois professores restantes, um deles usou no passado um jogo educativo para o ensino de matemática, mas o deixou de fazer por perder o acesso ao jogo, e o outro nunca chegou a cogitar jogos educativos por conta da matéria lecionada.

Todos os professores que cooperaram na pesquisa, contudo, têm uma opinião positiva acerca do uso de jogos em sala de aula, e em grande maioria possuem uma noção pessoal de como jogos podem ser usados como ferramenta de ensino. Em uma pergunta opcional onde perguntamos como que o professor em questão acredita que jogos podem ser bons recursos didáticos, quase todos os onze deram uma resposta. As respostas de cada professor se encontram no quadro abaixo.

¹⁰ “Ciências”, neste contexto, se refere às áreas, tanto do Ensino Fundamental quanto do Ensino Médio, de Ciências da Natureza conforme a BNCC homologada em 2018

Quadro 5: Justificativas para a pergunta “Você considera jogos educativos bons recursos didáticos?”

Qual é a faixa etária dos seus alunos?	Quais matérias você leciona para seus alunos?	Você considera jogos educativos bons recursos didáticos? Justifique, se possível:
menos que 5 anos	Matemática e português	Porque é mais fácil aprender com o concreto do que só com a teoria
entre 5 e 6 anos (alfabetização)	Pensamento Computacional	Por serem uma forma lúdica de se conectar ao aluno.
entre 5 e 6 anos (alfabetização)	Português, Matemática, História, Geografia e Ciências	Torna mais dinâmica e lúdica a aprendizagem
entre 5 e 6 anos (alfabetização)	Todas	É uma forma de prender a atenção das crianças e ensinar de maneira lúdica os conteúdos
entre 7 e 10 anos (Ensino Fundamental I)	Português, Matemática, Geografia	
entre 11 e 14 anos (Ensino Fundamental II)	Matemática	Os Jogos Educativos engaja os alunos no processo de ensino-aprendizagem.
entre 11 e 14 anos (Ensino Fundamental II)	História	Porque estimulam pensamento, raciocínio lógico e interação entre os envolvidos.
entre 15 e 17 anos (Ensino Médio)	Matemática e Física	Os jogos educativos desenvolvem o pensamento lúdico e a velocidade de raciocínio, tanto lógico quanto matemático
entre 15 e 17 anos (Ensino Médio)	Química	Eles atraem a atenção dos alunos e despertam o interesse para o assunto abordado de forma lúdica.
entre 15 e 17 anos (Ensino Médio)	Matemática	
18 anos ou mais	Termodinâmica Clássica, Termodinâmica Aplicada, Condução de Calor, Radiação Térmica	Ajuda a despertar o raciocínio

Fonte: Acervo do autor (2020)

Dos professores que usam jogos educativos atualmente, alguns exemplos de jogos que são usados são os jogos eletrônicos *Minesweeper Genius*, *Lightbot: Code Hour*, *Code Kartz* e a linguagem de programação *Scratch Jr.*, mencionados pelo professor dedicado de PC. Outro professor, desta vez de matemática, disse fazer os próprios jogos e estimular alunos a fazerem

jogos também. Os outros professores, contudo, mencionaram usar jogos não-eletrônicos, como jogos da memória, blocos lógicos, quebra-cabeças, tabuadas e, no caso de alunos menores, bingo.

O único professor que alegou ter usado jogos no passado mencionou ter utilizado a plataforma Mangahigh, uma plataforma online com jogos educativos onde professores de uma instituição podem acompanhar o desempenho dos alunos. De acordo com o professor, o que o levou a parar de usar os jogos da plataforma foi a perda de acesso, uma vez que o Mangahigh é pago e exige que a instituição de ensino seja cadastrada no serviço.

4.3 DESCRIÇÃO DA PESQUISA DE VALIDAÇÃO

Nesta seção, descreveremos os aspectos da pesquisa conduzida para validar o trabalho. Na seção 4.3.1, falaremos sobre a pesquisa em linhas gerais, elucidando detalhes sobre as intenções da pesquisa e as ações que tomamos para esses fins. Na seção 4.3.2, detalhamos o formulário passado para os voluntários que aceitaram participar do projeto, explicando cada pergunta apresentada aos voluntários. Na seção 4.3.3, revelamos a nossa abordagem diante das respostas para extrairmos informações que pudéssemos usar para validar o trabalho e traçar metas futuras.

4.3.1 Descrição da pesquisa

A pesquisa conduzida trata-se de uma pesquisa qualitativa. Por este trabalho de conclusão de curso se tratar de um pedaço de mídia que dialoga com outros campos do conhecimento, e devido ao momento único em que a validação do trabalho foi conduzida, a pesquisa qualitativa se mostrou a única solução viável para o nosso esforço de validação deste trabalho.

A pesquisa teve como objetivos provar, em algum nível, que o jogo é capaz de auxiliar no ensino de um conteúdo específico, saber se o jogo em questão tem a capacidade de ser jogado por pessoas que não possuem familiaridade com conceitos de computação, tentar prever se o jogo seria aceito por crianças e adolescentes dentro da faixa etária pretendida,

analisar o quão desafiador o jogo é para alunos desta faixa etária e coletar *feedback* de pessoas de fora do trabalho de modo a tornar o jogo mais jogável.

De modo a atingir os objetivos da pesquisa, contato com os voluntários foi feito através de e-mail e redes sociais. O corpo da mensagem era composto de uma introdução breve dos alunos, um link para um formulário montado no Google Forms e uma explicação breve sobre a motivação por trás do contato. O formulário inicia com uma explicação breve sobre os principais conceitos de pensamento computacional, para que os entrevistados possam julgar a efetividade do jogo em ensiná-los, e um termo de consentimento. Caso o entrevistado aceite o termo, são feitas perguntas acerca de sua experiência como professor, de modo a compreendermos seu perfil. Depois disso, o formulário fornece um *link* para o jogo e faz perguntas referentes à efetividade do jogo tanto como atividade lúdica quanto para como atividade educativa. Cerca de 19 professores foram contactados desta forma, e onze respostas foram coletadas.

4.3.2 Descrição do formulário de pesquisa

O voluntário, ao responder o formulário, passa sempre por cinco seções de perguntas.

A primeira seção é aberta com um texto curto reiterando o porquê da pesquisa e explicando brevemente sobre os princípios de PC. É nesta seção também que o voluntário aceita participar da pesquisa e ter as respostas fornecidas usadas neste trabalho. Como forma de identificação do voluntário, também pedimos o e-mail. Por fim, a primeira seção pergunta se o mesmo gostaria de receber os resultados da pesquisa uma vez que o formulário for fechado. Se o voluntário responder que não aceita ter os resultados usados na pesquisa, o formulário é enviado, registrando a recusa do voluntário em responder às perguntas.

A segunda seção é para coletar informações sobre o voluntário, para termos alguma noção do perfil do professor que está respondendo às perguntas. Pergunta-se, nesta seção, a faixa etária dos alunos para os quais o professor leciona — através de *checkboxes* com faixas etárias de acordo com as etapas da Educação Básica (BNCC, 2018) — as matérias às quais o professor leciona, sua opinião acerca de jogos didáticos e uma justificativa opcional, e por fim qual é a experiência pregressa do professor com jogos educativos em sala de aula.

Professores que usam jogos educativos no momento da resposta são perguntados sobre quais jogos e qual é a frequência com a qual eles usam jogos educativos em sala de aula. Professores que usaram jogos educativos em algum momento passado, mas por algum motivo ao momento da pesquisa não usam, foram perguntados sobre quais jogos eram usados e quais foram os motivos para terem parado de usar jogos em sala de aula. Professores que nunca usaram, mas gostariam de usar, foram perguntados sobre o motivo pelo qual jogos nunca foram utilizados. Por fim, professores que nunca consideraram o uso foram perguntados sobre o porquê de nunca terem pensado sobre.

A ideia no que diz respeito às seções acima é, a partir do *feedback* que será passado nas próximas perguntas, compreender se, na percepção pessoal de professores com experiência ao lidarem com diferentes alunos, o jogo é apropriado para as faixas etárias. No caso, apropriado tanto enquanto uma atividade de lazer quanto uma possível forma de ensinar alguma coisa. Todas as perguntas nas seções seguintes partem do pressuposto que o professor jogou o suficiente do jogo para ter uma opinião formada sobre o mesmo.

Na próxima seção, os voluntários se deparam com perguntas sobre os aspectos lúdicos do jogo. A página da seção tem link para o jogo e um breve texto explicando sobre as intenções do jogo, tanto sobre o público alvo pretendido quanto os conhecimentos que o jogo visa reforçar. Como não podemos assistir aos professores jogando para termos uma noção do quanto foi jogado, a primeira pergunta da seção é a questão “Até que parte do jogo você jogou”. Em seguida, perguntamos se o professor acha que os alunos dele gostarão de jogar o jogo. Então perguntamos se o voluntário acha que os alunos se sentirão desafiados pelas fases do jogo. Depois, perguntamos se o jogo é apropriado para a faixa etária. Por fim, perguntamos se o voluntário desgostou de algum aspecto do jogo, dando um campo de texto opcional em seguida para o professor se alongar caso tenha achado algo que ele veio a desgostar.

Na próxima e última seção, os voluntários se deparam com perguntas sobre os aspectos pedagógicos do jogo. A página da seção tem link para o jogo e um breve texto explicando sobre as intenções do jogo, tanto sobre o público alvo pretendido quanto os conhecimentos que o jogo visa reforçar. Todas as perguntas são obrigatórias para completar o formulário. A seção começa perguntando como que o voluntário acredita que o jogo ajudaria no aprendizado dos alunos dele. A próxima pergunta pede para o voluntário avaliar, usando a escala de Likert, o quanto que o jogo consegue ensinar os quatro pilares de PC para os jogadores. Em seguida, pergunta-se ao voluntário se ele acredita que o jogo pode prejudicar o

aprendizado de alguma forma. Depois, o formulário pergunta se o voluntário mudaria alguma coisa no jogo. Por fim, se o voluntário tiver alguma experiência pregressa usando jogos em sala de aula, o formulário pergunta se o jogo deste trabalho tem um diferencial para com outros jogos sobre pensamento computacional. Caso o voluntário tenha respondido anteriormente nunca ter usado jogos, contudo, o formulário não mostra a última pergunta mencionada.

4.3.3 Tratamento das respostas

Em um primeiro momento, as respostas foram todas compiladas em uma planilha no *Google Sheets* gerada pelo próprio *Google Forms*, mostrando todas as respostas dadas pelos voluntários em uma planilha com quarenta e duas colunas. Trabalhar com uma planilha tão grande torna a visualização de dados para interpretação muito tortuosa, então quebramos essa planilha grande em pequenas planilhas mais legíveis.

Ao todo, a planilha de resultados originais foi dividida em dez subplanilhas para maior legibilidade. Todas as sub-planilhas foram copiadas da grande planilha, de modo que, caso algum voluntário responda ao formulário, ou alguma alteração às respostas sejam feitas, as planilhas menores serão automaticamente atualizadas.

A primeira sub-planilha foi criada para uma para contar o número de professores com experiência em lecionar para uma certa faixa etária de alunos. Essa planilha gerou o Gráfico 1.

A segunda sub-planilha conta o número de professores que lecionam matérias específicas. Se um voluntário respondeu lecionar mais de uma matéria, ele entra na contagem de todas as matérias para as quais dá aula. Ao todo, o número de professores que leciona uma matéria específica não deve ser maior do que o total de voluntários que responderam ao questionário. É dessa planilha de onde saiu o Gráfico 2.

A terceira sub-planilha reúne as opiniões de cada voluntário acerca do uso de jogos em sala de aula como atividade lúdica. Sua função, além de agregar as opiniões dos voluntários, é correlacionar essas opiniões com as matérias e a faixa etária para as quais o professor em questão leciona.

A quarta sub-planilha contabiliza quantos voluntários possuem qual relação com jogos educativos em sala de aula. A partir dela, não só se tem a contagem absoluta de voluntários (por exemplo, o número exato de professores que usam jogos educativos) como também a proporção de cada resposta levando em consideração as respostas dos outros voluntários.

A quinta sub-planilha possui apenas as respostas de voluntários que usam jogos educativos no momento da pesquisa às perguntas específicas a este grupo. A sexta sub-planilha, por sua vez, possui apenas as respostas de voluntários que usaram jogos em algum momento do passado deram às perguntas específicas para este grupo. Já a sétima sub-planilha guarda apenas as respostas de voluntários que nunca usaram jogos, mas gostariam de usar, às perguntas específicas para este grupo. Por fim, a oitava sub-planilha contém as respostas dos voluntários que nunca consideraram o uso de jogos às perguntas específicas para esse grupo.

A antepenúltima sub-planilha é onde estão compiladas as respostas dos voluntários acerca do jogo enquanto atividade lúdica para seus alunos. A partir desta seção, geramos gráficos para avaliarmos o percentual de voluntários que acreditam que o jogo será bem-recebido pelos seus alunos, até onde cada voluntário jogou o jogo e a relação entre a faixa etária dos alunos de cada voluntário e as opiniões de cada voluntário tanto sobre o jogo ser apropriado para a faixa etária de seus alunos quanto sobre o jogo possivelmente ser bem recebido por esses mesmos alunos. Esta sub-planilha também se utilizou de formatação condicional para colorir as células de acordo com as respostas dos voluntários para facilitar a visualização.

A penúltima sub-planilha é onde estão compiladas as respostas dos voluntários acerca do jogo enquanto um artefato didático para ensinar PC. O único gráfico gerado a partir desta sub-planilha foi a relação entre os pilares de PC existentes e a opinião dos voluntários dentro da escala de Likert sobre o quão bem o jogo ensina cada pilar.

As duas sub-planilhas acima são as mais importantes porque a legibilidade da planilha original é em maior parte comprometida pela forma como o *Google Forms* organiza as páginas que os voluntários acessam ao responder às perguntas, fazendo com que criássemos duplicatas das perguntas das últimas seções para voluntários familiarizados com o uso de jogos educativos em sala e voluntários sem familiaridade, de modo a podermos fazer uma pergunta adicional para voluntários familiarizados.

A última sub-planilha foi montada durante discussões diante da necessidade de relacionar rapidamente respostas desenvolvidas em texto com características de cada voluntário. Dado o número pequeno de respostas, montar essa planilha foi viável. Ao final, montamos uma planilha relacionando a faixa etária de cada voluntário, as matérias que cada voluntário leciona, e quatro respostas em texto que cada voluntário deu sobre o jogo apresentado.

4.4 RESULTADOS DA PESQUISA

Nesta seção, interpretaremos as respostas dadas acerca do jogo através da pesquisa, tentando sempre relacionar o perfil dos voluntários e de seus alunos às respostas oferecidas e tirando conclusões a partir dessa relação dentro dos limites do razoável. Na seção 4.4.1, relacionaremos a faixa etária dos alunos e as matérias lecionadas pelos voluntários às opiniões expressadas sobre se o jogo era apropriado ou atende aos gostos dos alunos. Na seção 4.4.2, fazemos a mesma relação entre faixa etária e matéria lecionada, só que desta vez com o parecer de cada voluntário acerca do trabalho que o jogo faz tentando ensinar cada pilar de PC. Por fim, na seção 4.4.3, discorreremos sobre a opinião de cada voluntário sobre como o jogo ajuda no aprendizado de PC, procurando por padrões nas respostas.

4.4.1 Opinião acerca da aceitação do jogo por parte dos alunos

As primeiras questões para as quais olhamos na hora de avaliar as respostas do voluntários foram saber se estes consideram o nosso jogo apropriado para os alunos e se acreditam que seus alunos gostariam dele. Nossa preocupação seria que o jogo, apesar de nosso estudo, não fosse considerado apropriado ou atraente por figuras que lidam com o nosso público-alvo. Por causa da natureza dos voluntários, procuramos também testar como professores que dão aulas para alunos em faixas etárias diferentes da pretendida percebem o jogo.

Nesta subseção, analisaremos em maior parte duas perguntas: “Este jogo é apropriado para a faixa etária dos seus alunos?” e “Seus alunos gostarão de jogar este jogo?”. O quadro e

os gráficos abaixo sintetizam as respostas para ambas as perguntas, mostrando os resultados tanto em números absolutos quanto em percentuais.

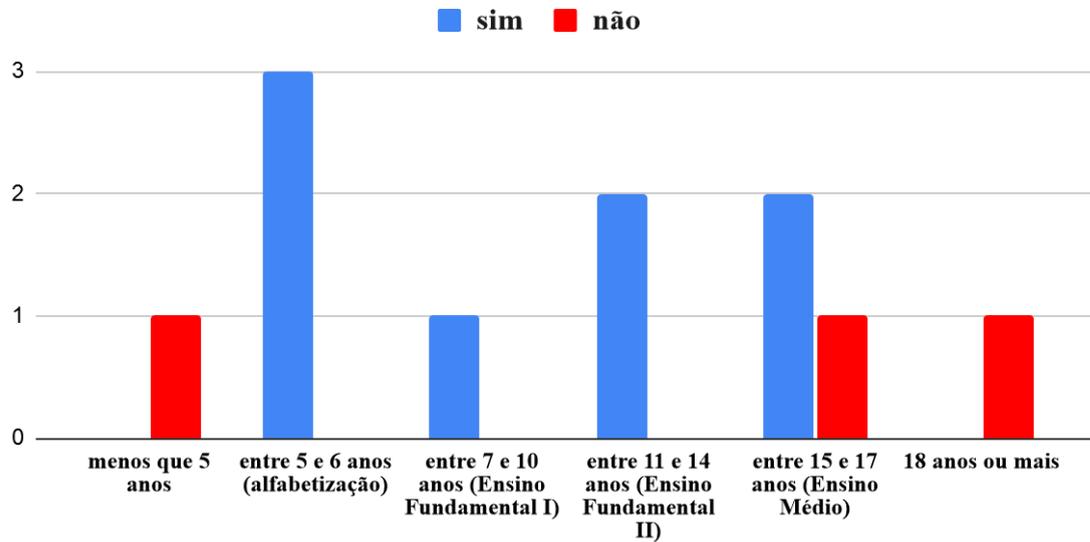
Quadro 6: Relação entre faixa etária, matéria lecionada e opiniões sobre o jogo

FAIXA ETÁRIA	MATÉRIA LECIONADA	ESTE JOGO É APROPRIADO PARA OS SEUS ALUNOS?	OS SEUS ALUNOS GOSTARÃO DE JOGAR ESTE JOGO?
menos que 5 anos	Matemática e português	Não	Não
entre 5 e 6 anos (alfabetização)	Pensamento Computacional	<i>Sim</i>	<i>Sim</i>
entre 5 e 6 anos (alfabetização)	Português, Matemática, História, Geografia e Ciências	<i>Sim</i>	<i>Sim</i>
entre 5 e 6 anos (alfabetização)	Todas	<i>Sim</i>	<i>Sim</i>
entre 7 e 10 anos (Ensino Fundamental I)	Português, Matemática, Geografia	<i>Sim</i>	<i>Sim</i>
entre 11 e 14 anos (Ensino Fundamental II)	História	<i>Sim</i>	<i>Sim</i>
entre 11 e 14 anos (Ensino Fundamental II)	Matemática	<i>Sim</i>	<i>Sim</i>
entre 15 e 17 anos (Ensino Médio)	MATEMÁTICA E FÍSICA	<i>Sim</i>	<i>Sim</i>
entre 15 e 17 anos (Ensino Médio)	Química	Não	<i>Sim</i>
entre 15 e 17 anos (Ensino Médio)	Matemática	<i>Sim</i>	Não
18 anos ou mais	Termodinâmica Clássica, Termodinâmica Aplicada, Condução de Calor, Radiação Térmica	Não	Não

Fonte: Acervo do autor (2020)

Gráfico 2: Resultados absolutos da pergunta “Este jogo é apropriado para a faixa etária dos seus alunos?”

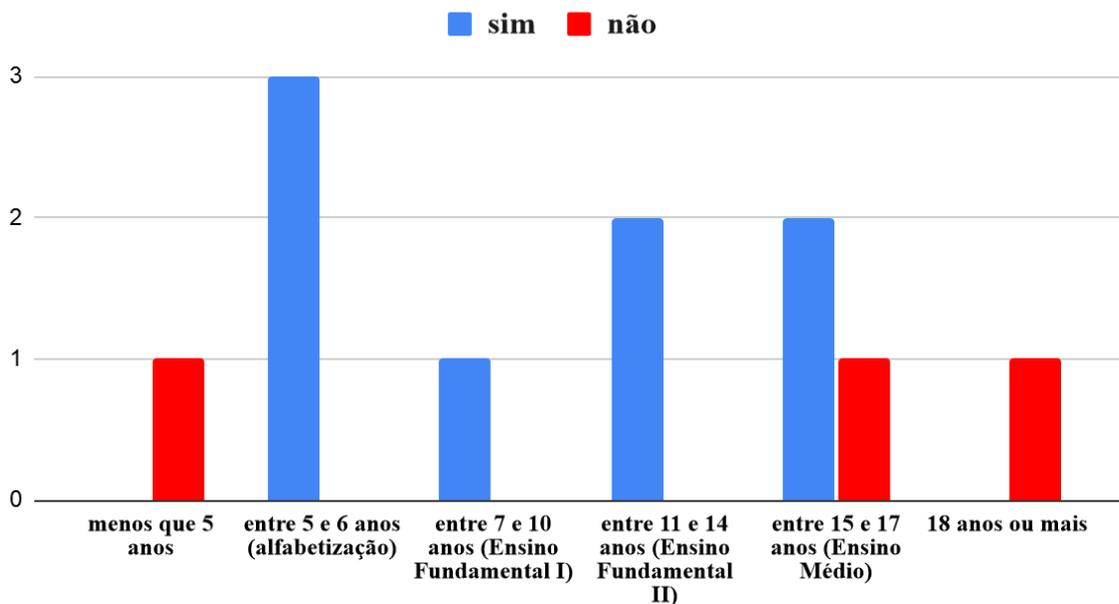
Este jogo é apropriado para a faixa etária dos seus alunos?



Fonte: Acervo do autor (2020)

Gráfico 3: Resultados absolutos da pergunta “Seus alunos gostarão de jogar este jogo?”

Seus alunos gostarão de jogar este jogo?



Fonte: Acervo do autor (2020)

Os resultados apontaram para um cenário onde voluntários que dão aulas para crianças muito pequenas e pessoas com mais de dezoito anos consideram o jogo não apropriado para seus alunos, prevendo também que esses mesmos alunos virão a não gostar do jogo. Enquanto isso, voluntários com alunos na faixa etária dos cinco aos catorze anos concordaram que o jogo seria tanto apropriado quanto bem-recebido pelos seus alunos. O único conflito vem quando se analisa a faixa etária dos quinze aos dezessete anos, onde uma maioria dos voluntários consideram o jogo apropriado e do gosto de seus alunos.

Dentre os voluntários que dão aula para alunos abaixo da faixa etária proposta, consta um professor dedicado de PC familiarizado com o uso de jogos educativos em sala que respondeu que o jogo é apropriado e do gosto de seus alunos. Esse voluntário específico respondeu que a faixa etária de seus alunos está entre cinco e sete anos de idade.

Acerca dos voluntários que lecionam para alunos de quinze a dezessete anos, uma das respostas aponta que os alunos gostarão do jogo, embora este não seja apropriado para os seus alunos. Outra resposta, contudo, acusa o contrário; isto é, embora o jogo seja apropriado para os seus alunos, os mesmos não viriam a gostar do jogo. No caso, os dois voluntários se pronunciaram sobre aspectos que eles mesmos desgostam, mas isso será coberto na seção 4.5.

Esses resultados nos serviram de alguma prova de que o nosso trabalho é visto por educadores como apropriado e próprio para crianças e adolescentes dentro da nossa faixa etária pretendida e até de outras, com divergências de opiniões surgindo conforme nos afastamos da faixa etária pretendida de sete a onze anos.

4.4.2 Pilares de pensamento computacional

Uma vez provado que o jogo pode ser do gosto do nosso público-alvo e que o jogo pode ser jogado por pessoas sem familiaridade com jogos de modo geral, focamos em avaliar a efetividade do jogo em ajudar no ensino de PC. De modo a analisar isso, perguntamos aos voluntários o quão bem eles acreditam que cada pilar foi exercitado através das fases do jogo. É importante levar em consideração, contudo, que PC não é a área de expertise de todos os voluntários que responderam ao questionário, visto que apenas um deles é um professor dedicado de PC.

Nesta subseção, analisaremos as perguntas “Você concorda que o jogo consegue ensinar os seguintes pilares de pensamento computacional?”, onde apresentamos os quatro pilares documentados e uma escala de Likert. O quadro e os gráficos abaixo sintetizam as respostas de cada voluntário para essa questão, mostrando os resultados tanto em números absolutos quanto percentuais.

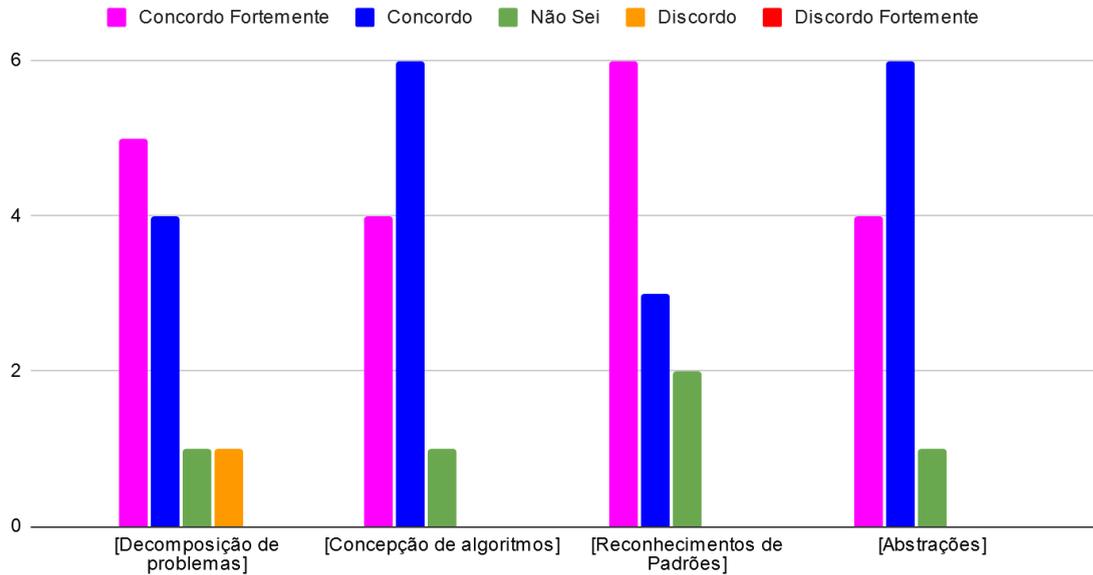
Quadro 7: Relação entre faixa etária, matéria lecionada, até qual grupo de fases o voluntário jogou e respostas à pergunta “Você concorda que o jogo consegue ensinar os seguintes pilares de pensamento computacional?”

Qual é a faixa etária dos seus alunos?	Quais matérias você leciona para seus alunos?	Até que parte do jogo você jogou?	Decomposição de problemas	Concepção de algoritmos	Reconhecimento de Padrões	Abstrações
menos que 5 anos	Matemática e português	2: Empurrar	Discordo	Concordo fortemente	Concordo fortemente	Não sei
entre 5 e 6 anos	Pensamento Computacional	1: Andar	Não sei	Concordo fortemente	Não sei	Concordo
entre 5 e 6 anos	Português, Matemática, História, Geografia e Ciências	4: Árvores	Concordo	Concordo	Não sei	Concordo
entre 5 e 6 anos	Todas	6: Concorrência	Concordo fortemente	Concordo	Concordo fortemente	Concordo
entre 7 e 10 anos	Português, Matemática, Geografia	1: Andar	Concordo	Concordo	Concordo	Concordo
entre 11 e 14 anos	Matemática	(Sem resposta)	Concordo fortemente	Concordo fortemente	Concordo fortemente	Concordo fortemente
entre 11 e 14 anos	História	5: Laços	Concordo	Não sei	Concordo	Concordo
entre 15 e 17 anos	MATEMÁTICA E FÍSICA	6: Concorrência	Concordo fortemente	Concordo fortemente	Concordo fortemente	Concordo fortemente
entre 15 e 17 anos	Química	5: Laços	Concordo fortemente	Concordo	Concordo fortemente	Concordo fortemente
entre 15 e 17 anos	Matemática	4: Árvores	Concordo	Concordo	Concordo	Concordo
18 anos ou mais	Termodinâmica Clássica, Termodinâmica Aplicada, Condução de Calor, Radiação Térmica	3: Pegar & Largar	Concordo fortemente	Concordo	Concordo fortemente	Concordo fortemente

Fonte: Acervo do autor (2020)

Gráfico 4: Resultados absolutos da pergunta “Você concorda que o jogo consegue ensinar os seguintes pilares de pensamento computacional?”

Você concorda que o jogo consegue ensinar os seguintes pilares de pensamento computacional?



Fonte: Acervo do autor (2020)

As respostas foram, de modo geral, positivas, afirmando que o jogo exercita tanto os pilares que eram o objetivo original do trabalho quanto pilares que não. Contudo, é importante destacar que, como PC não é a área de expertise da maioria dos nossos voluntários, nos levando inclusive a anexar no questionário uma breve explicação sobre o assunto, muitos votos podem ter sido dados sem que o voluntário compreenda plenamente a ideia por trás de cada pilar. Também é importante destacar um problema inerente da escala de Likert, que é a tendência de receber respostas positivas por parte dos voluntários (BERTRAM, 2016).

Ao mesmo tempo em que muitos voluntários não possuem familiaridade com o conteúdo, o voluntário específico que dá aula de PC para crianças expressou concordar fortemente com a noção do jogo ensinar dois dos quatro pilares, o que nos deixa mais seguros dos nossos esforços. Os pilares no caso são concepção de algoritmos, que foi nosso maior foco originalmente, e abstrações. Esses pilares também foram os que receberam maior número de votos concordantes, com apenas uma pessoa votando “não sei” em ambos pilares.

O único voluntário que explicitamente discordou da noção do jogo ensinar um dos pilares o fez em relação ao pilar de Decomposição de Problemas. Este voluntário também

concordou fortemente com a capacidade do jogo ensinar reconhecimento de padrões e concepção de algoritmos, e marcou não saber se o jogo efetivamente exercita capacidade de abstração. Esse mesmo voluntário é o único que dá aula para crianças com idades abaixo dos cinco anos de idade, e jogou até o segundo grupo de fases.

4.4.3 Como o jogo ajuda no aprendizado?

Uma vez provado que o jogo possui o potencial de ser do gosto de jovens da nossa faixa etária e de ser capaz de ensinar fundamentos de PC, nós questionamos os voluntários como estes acreditam que o jogo em questão pode agregar à educação de seus alunos. Como os voluntários que responderam ao questionário são de diversas áreas e possuem alunos de diversas idades, foi de nosso interesse compreender a visão individual de cada um acerca do uso pedagógico não só de jogos em geral, como do nosso jogo em específico.

Todos os voluntários que responderam ao questionário se mostraram positivos ao uso de jogos para fins didáticos, até mesmo aqueles que nunca usaram. A pergunta a ser analisada nesta seção é “Como você acredita que este jogo ajudaria no aprendizado dos seus alunos?”, e foi uma das perguntas que pediam por uma resposta escrita em texto corrido.

Quadro 8: Relação entre faixa etária dos alunos, matéria lecionada e respostas à pergunta “Como você acredita que este jogo ajudaria no aprendizado dos seus alunos?”

Qual é a faixa etária dos seus alunos?	Quais matérias você leciona para seus alunos?	Como você acredita que este jogo ajudaria no aprendizado dos seus alunos?
menos que 5 anos	Matemática e português	Noções de coordenadas (frente, lado)
entre 5 e 6 anos	Pensamento Computacional	Principalmente na parte de pensamento algorítmico
entre 5 e 6 anos	Português, Matemática.História , Geografia e Ciências	Estimularia o raciocínio lógico
entre 5 e 6 anos	Todas	No desenvolvimento cognitivo e raciocínio lógico
entre 7 e 10 anos	Português, Matemática,	Sim, jogos sempre são muito estimulantes.

	Geografia	
entre 11 e 14 anos	Matemática	Acredito que o mesmo possa colaborar com o Raciocínio Lógico.
entre 11 e 14 anos	História	Certamente estimularia o trabalho em equipe, se sentiriam desafiados a superarem suas limitações, com isso seu raciocínio seria bem mais estimulado, o que com certeza refletiria em outras atividades escolares.
entre 15 e 17 anos	Matemática e Física	Todo jogo que desenvolva esses tipos de raciocínios, são válidos
entre 15 e 17 anos	Química	Raciocínio lógico
entre 15 e 17 anos	Matemática	sem resposta
18 anos ou mais	Termodinâmica Clássica, Termodinâmica Aplicada, Condução de Calor, Radiação Térmica	O jogo é interessante, mas está fora da faixa etária dos meus alunos

Fonte: Acervo do autor (2020)

Nem todas as respostas dadas pelos voluntários atendem à questão levantada pela pergunta. Então nós vamos focar nas respostas que respondem à pergunta de forma apropriada, assim ignorando respostas que não respondem à questão.

As respostas apropriadas apontaram em sua maioria para um “desenvolvimento do raciocínio lógico”. Contudo, algumas respostas se destacam, sendo a primeira dada pelo voluntário que dá aula de história, que destacou o potencial do jogo de estimular o trabalho em equipe para resolver as fases, o que “refletiria em outras atividades escolares”. Essa visão dialoga com o princípio Vygotskyano de *scaffolding*, no qual, em um ambiente de ensino, alunos mais experientes servem de mentores ao elevarem alunos menos experientes. No caso do jogo, isso seria refletido no ato de jogadores mais experientes ajudando jogadores menos experientes a passarem de fases; um cenário que a gente não toma como garantia que seja a norma, mas que levamos em consideração ser uma possibilidade.

Outra resposta que se destacou foi a do voluntário que dá aula para crianças muito pequenas. Diferente dos outros voluntários com alunos mais velhos, a contribuição que essa resposta reconhece é bem mais simples e condizente com a faixa etária dos alunos. Por outro lado, confirmando a informação que a gente já assumiu através das respostas passadas, o

voluntário que dá aula para adultos reconheceu através desta questão a inadequação do jogo para com a faixa etária dos alunos dele.

Por fim, o voluntário que dá aula de PC respondeu que o jogo ajudaria com pensamento algorítmico, fazendo assim menção ao pilar de PC de concepção de algoritmos. Essa resposta nos deu ainda mais segurança de que os nossos esforços, de conceber um jogo focado em estimular as noções de concepção de algoritmos e decomposição de problemas, foram bem-sucedidos, ao resultar em um jogo reconhecidamente útil em estimular o desenvolvimento de PC de forma orgânica.

4.5 CRÍTICAS E SUGESTÕES

Nesta seção, analisaremos as sugestões dadas acerca do jogo pelos voluntários com base nas críticas pessoais de modo a trabalhar em melhorias a fim de tornar o jogo mais atraente para o público alvo. Esta seção girará em torno de duas perguntas feitas no questionário passado aos voluntários: a primeira pergunta é “Caso o jogo tenha algo que você desgostou, fale mais sobre, por favor”, enquanto que a segunda pergunta é “Se você pudesse mudar qualquer coisa nesse jogo, o que você mudaria?”. Estas duas perguntas admitem resposta em texto corrido, e as respostas podem ser vistas nos quadros 9 e 10 abaixo.

Quadro 9: Respostas da pergunta do formulário “Caso o jogo tenha algo que você desgostou, fale mais sobre, por favor”

Qual é a faixa etária dos seus alunos?	Quais matérias você leciona para seus alunos?	Caso o jogo tenha algo que você desgostou, fale mais sobre, por favor
menos que 5 anos	Matemática e português	Achei complicado ter que colocar todas as setas ali do lado, ao invés de simplesmente clicar nas setas e o boneco se mexer. Também encontrei problema com o tutorial, porque não conseguia sair dele. Não consegui avançar da fase dois porque ficava entrando no tutorial e eu não conseguia jogar a fase.
entre 5 e 6 anos (alfabetização)	Pensamento Computacional	
entre 5 e 6 anos (alfabetização)	Português, Matemática.História, Geografia e Ciências	Em alguns casos , a explicação não foi clara.
entre 5 e 6 anos	Todas	Acredito que crianças muito pequenas podem achar o sistema de

(alfabetização)		setas um pouco confuso
entre 7 e 10 anos (Ensino Fundamental I)	Português, Matemática, Geografia	
entre 11 e 14 anos (Ensino Fundamental II)	Matemática	Ele me pareceu um pouco sugado, como na fase pra pegar o cachorro. Dizia que eu havia batido na parede, mas não tinha parede.
entre 11 e 14 anos (Ensino Fundamental II)	História	Eu penso que as instruções não estavam bem esclarecidas, principalmente na etapa em que eu parei, a dos 'Laços'.
entre 15 e 17 anos (Ensino Médio)	Matemática e Física	Nada a declarar
entre 15 e 17 anos (Ensino Médio)	Química	Atuo no ensino médio. Minha sugestão seria que o jogo tivesse uma história para atrair mais alunos de idades mais avançadas. Assim o objetivo não seria apenas ir de um ponto ao outro e sim concluir o objetivo para desvendar o que acontece com determinado personagem.
entre 15 e 17 anos (Ensino Médio)	Matemática	Se o jogo for realizado no PC, o fato de arrastar os movimentos necessários, deixa o jogo com pouca habilidade.
18 anos ou mais	Termodinâmica Clássica, Termodinâmica Aplicada, Condução de Calor, Radiação Térmica	

Fonte: Acervo do autor (2020)

Três aspectos do jogo foram continuamente apontados como falhos: os tutoriais, responsáveis por darem a base para que o jogador possa jogar as fases do jogo, o modo como dar instruções aos personagens na tela e a resolução gráfica que ativamente impediu pelo menos um voluntário de continuar a jogar. Outras respostas à pergunta fizeram menção à falta de feedback para as ações tomadas, falta de um enredo para gerar engajamento emocional e preocupação com “crianças muito pequenas” tendo dificuldade em lidar com as instruções na forma de setas. Esta última preocupação, apesar de válida, é a que menos nos preocupa, uma vez que o nosso foco é realmente uma faixa etária maior, em um estágio mais avançado do desenvolvimento da cognição.

Curiosamente, nenhuma crítica feita pelos voluntários teve relação com a falta de efeitos sonoros ou música ambiente.

Quadro 10: Respostas da pergunta do formulário “Se você pudesse mudar qualquer coisa nesse jogo, o que você mudaria?”

Qual é a faixa etária dos seus alunos?	Quais matérias você leciona para seus alunos?	Se você pudesse mudar qualquer coisa nesse jogo, o que você mudaria?
menos que 5 anos	Matemática e português	Modo de andar com o boneco.
entre 5 e 6 anos (alfabetização)	Pensamento Computacional	Não saberia dizer nesse momento.
entre 5 e 6 anos (alfabetização)	Português, Matemática, História, Geografia e Ciências	Em algumas etapas tive que jogar para entender. O tutorial não deixou claro o que deveria ser feito.
entre 5 e 6 anos (alfabetização)	Todas	Nada
entre 7 e 10 anos (Ensino Fundamental I)	Português, Matemática, Geografia	Nada
entre 11 e 14 anos (Ensino Fundamental II)	Matemática	Talvez uma tela inicial que pudesse direcionar melhor a idade do aluno.
entre 11 e 14 anos (Ensino Fundamental II)	História	Provavelmente eu melhoraria a parte sobre instruções, os tutoriais. De resto está bem configurado.
entre 15 e 17 anos (Ensino Médio)	Matemática e Física	Nada
entre 15 e 17 anos (Ensino Médio)	Química	Adicionaria um enredo ao jogo para tornar mais atrativo.
entre 15 e 17 anos (Ensino Médio)	Matemática	a movimentação dos personagens
18 anos ou mais	Termodinâmica Clássica, Termodinâmica Aplicada, Condução de Calor, Radiação Térmica	A resolução gráfica

Fonte: Acervo do autor (2020)

As respostas a esta pergunta muito refletem das críticas feitas no quadro 9. As alterações mais citadas foram o modo como dar instruções aos personagens e os tutoriais das fases, de modo a clarificar não só como interagir com mecânicas e dinâmicas do jogo como também os objetivos das fases. Curiosamente, o voluntário que apontou preocupação com

crianças muito pequenas não expressou nenhuma vontade de mudar nada no jogo, e um voluntário que não expressou desgosto por nenhum aspecto do jogo relatou que mudaria a resolução gráfica, sem entrar em maiores detalhes.

Um último detalhe é o voluntário que dá aula de PC para crianças não expressando nenhum desgosto ou desejo de mudança. Esse detalhe não invalida nenhuma das observações feitas por outros voluntários, mas essa resposta nos tranquiliza, uma vez que alguém com *expertise* em PC não sente que o nosso jogo seja impróprio, nem tenha vontade de mudar nenhum aspecto dele.

5 CONCLUSÃO

Este trabalho consistiu na criação de um jogo educativo com o propósito de ajudar no ensino de PC para crianças e adolescentes. A faixa etária visada para o jogo são jovens entre sete e onze anos, sendo assim demarcada com base na teoria piagetiana de desenvolvimento da cognição. O jogo possui, no seu estado atual, vinte e duas fases divididas em seis grupos que introduzem mecânicas relacionadas a conteúdos pedagógicos retirados de currículos de computação usadas por escolas com ensino formal de programação para jovens de faixas etárias próximas. Desta forma, o jogo visa transmitir ao jogador a capacidade de usar dois dos quatro pilares que fundamentam o que se compreende hoje em dia como PC: a decomposição de problemas em problemas menores e menos complexos e a concepção de algoritmos para resolução de problemas. Toda a estética do jogo ao redor gira em torno de recompensar o jogador por aplicar ditos conhecimentos mas sem criar pressão que possa dificultar o ato de pensar em soluções para os problemas propostos pelas fases.

De modo a validarmos o jogo como algo que pode ser usado para ajudar no ensino de jovens, procuramos entrar em contato com professores que estivessem dispostos a jogar o jogo e dar seus pareceres sobre se o jogo seria de interesse de seus alunos ou não. Como o trabalho foi conduzido em meio a uma pandemia, o contato com professores e o controle sobre as condições em que o jogo foi jogado foi limitado. A priori, apenas professores de PC, computação ou matemática seriam consultados, mas esse requisito foi repensado de modo a termos um número suficiente de respostas e contemplarmos pareceres de professores de outras áreas. Ao final da pesquisa, onze professores foram consultados e tiveram suas respostas consideradas.

A validação se deu através de uma pesquisa qualitativa feita através da internet, enviando o jogo disponível para computadores e celulares *Android* e um questionário com perguntas acerca do professor e do seu parecer acerca do jogo. A recepção foi positiva e nos reafirmou que o jogo é apropriado enquanto jogo e enquanto facilitador no ensino de PC, com as maiores críticas sendo no que diz respeito a problemas técnicos ou questões de polimento, com alguns voluntários mais distantes da área tendo dificuldade em compreender como interagir com o jogo ou cumprir objetivos.

5.1 TRABALHOS FUTUROS

Apesar da boa recepção, ainda existe trabalho para ser feito. O jogo, na sua forma atual, ainda apresenta o problema onde o *overlay* da plataforma *itch.io* pode cobrir botões essenciais para se jogar o jogo em computadores. Além disso, o jogo não existe na plataforma *PlayStore* para fácil instalação em celulares *Android*, exigindo que um possível jogador passe por mais passos para poder instalar o jogo no celular. Também, de acordo com as críticas recebidas, seria bem-vindo retrabalhar os tutoriais de cada grupo de fase para que esclareçam melhor não só como as mecânicas dentro do jogo funcionam, como também como lidar com as mesmas.

Sugestões recebidas também apontaram para a criação de um enredo de modo a engajar emocionalmente os jogadores mais velhos, e para a elaboração de uma interface mais apropriada para computadores. Também foi apontado em uma resposta específica da pesquisa que mais *feedback* acerca das ações tomadas pelos personagens seria bem-vindo, falando que “Dizia que eu havia batido na parede, mas não tinha parede.”. Este problema seria resolvido se o jogo tivesse *feedback* visual e sonoro para quando um personagem bate na parede, ao invés de apenas interromper a execução do algoritmo e lançar a mensagem de erro. .

Outra ideia de trabalho futuro é a criação de mais fases, com a possibilidade de haver fases que permitam mais do que cinco instruções por personagem e/ou tenham um cenário maior. Essas novas fases também poderão introduzir outros desafios para serem resolvidos de forma algorítmica e outros personagens. Também é possível criar um modo “infinito”, onde fases novas são geradas proceduralmente conforme o jogador as conclui, e um modo criativo para o jogador poder explorar as mecânicas do jogo e criar suas próprias fases, de modo a poder compartilhá-las com o mundo.

Por fim, após estudarmos outros jogos similares citados na seção 2.6, percebemos que a maior parte desses jogos oferecem um artefato que o nosso não possui até agora: um *scoreboard* para comparar os algoritmos de um jogador aos algoritmos de outros jogadores. Jogos como *Opus Magnum* e *TIS-100* comparam a solução do jogador com as soluções de outros jogadores que constem na lista de “amigos” do jogador, inserindo no jogo um elemento de competitividade entre jogadores para a construção do algoritmo mais eficiente e compacto, ao mesmo tempo em que faz alusão à possibilidade de existirem soluções melhores para um

determinado problema. Esse detalhe pode estimular o jogador a jogar uma fase de novo atrás de um desempenho melhor.

REFERÊNCIAS

- ABBOTT, T. **MDA Framework- Unconnected Connectivity**. Disponível em: https://www.gamasutra.com/blogs/TuckerAbbott/20101212/88611/MDA_Framework_Unconnected_Connectivity.php. Acesso em: 2 jul. 2020.
- BARR, D.; HARRISON, J.; CONERY, L. Computational Thinking: A Digital Age Skill for Everyone. **Learning & Leading with Technology**, p. 20–23, abr. 2011.
- BARR, V.; STEPHENSON, C. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? **ACM Inroads**, v. 2, n. 1, p. 48–54, 25 fev. 2011.
- BERTRAM, D. **Likert Scales**. p. 11, 2016.
- BRACKMANN, C. et al. **Computational thinking: Panorama of the Americas**. 2016 International Symposium on Computers in Education (SIIE). **Anais...** In: 2016 INTERNATIONAL SYMPOSIUM ON COMPUTERS IN EDUCATION (SIIE). Salamanca, Spain: IEEE, set. 2016Disponível em: <http://ieeexplore.ieee.org/document/7751839/>. Acesso em: 6 jun. 2020
- BRASIL. Ministério da Educação. Base Nacional Comum Curricular. Brasília, 2018.
- DE FRANÇA, R. S. et al. **A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação**. 10, 2014.
- DJAOUTI, Damien; ALVAREZ, Julian; JESSEL, Jean-Pierre. **Classifying Serious Games: the G/P/S model**. Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches. 2011. DOI: <http://dx.doi.org/10.4018/978-1-60960-495-0.ch006>.
- FERRARI, M. **Jean Piaget, o biólogo que colocou a aprendizagem no microscópio**. Disponível em: <https://novaescola.org.br/conteudo/1709/jean-piaget-o-biologo-que-colocou-a-aprendizagem-no-microscopio>. Acesso em: 18 jun. 2020a.

FERRARI, M. **Lev Vygotsky: o teórico do ensino como processo social**. Disponível em: <https://novaescola.org.br/conteudo/382/lev-vygotsky-o-teorico-do-ensino-como-processo-social>. Acesso em: 18 jun. 2020b.

GROVER, S.; PEA, R. Computational Thinking in K–12: A Review of the State of the Field. **Educational Researcher**, v. 42, n. 1, p. 38–43, jan. 2013.

HALVERSON, R. What Can K-12 School Leaders Learn from Video Games and Gaming? p. 8, [s.d.].

HUNICKE, R.; LEBLANC, M.; ZUBEK, R. MDA: A Formal Approach to Game Design and Game Research. p. 5, [s.d.].

JUUL, J. **Half-real: video games between real rules and fictional worlds**. Cambridge, Mass.: MIT Press, 2005.

KIRKPATRICK, K. Parallel computational thinking. **Communications of the ACM**, v. 60, n. 12, p. 17–19, 27 nov. 2017.

KRATHWOHL, D. R. A Revision of Bloom’s Taxonomy: An Overview. **Theory Into Practice**, v. 41, n. 4, p. 212–218, nov. 2002.

MCLEOD, S. **Jean Piaget’s Theory and Stages of Cognitive Development | Simply Psychology**. Disponível em: <https://www.simplypsychology.org/piaget.html#stages>. Acesso em: 18 jun. 2020.

MCLEOD, S. **Zone of Proximal Development and Scaffolding | Simply Psychology**. Disponível em: <https://www.simplypsychology.org/Zone-of-Proximal-Development.html>. Acesso em: 18 jun. 2020.

MONCLAR, R. Jogos com Propósito para o Ensino de Programação. p. 9, [s.d.].

MYERS, B. A. Taxonomies of visual programming and program visualization. **Journal of Visual Languages & Computing**, v. 1, n. 1, p. 97–123, mar. 1990.

MYERS, B. A. Visual Programming, Programming by Example and Program Visualization: A Taxonomy. [s.d.].

NASCIMENTO, Pedro Vitor Marques. **Mapa do Tesouro**. Disponível em: <https://tanookihouse.itch.io/mapa-do-tesouro>. Acesso em: 13 dez. 2020.

Operational Definition of Computational Thinking for K-12 Education. , 2011. Disponível em: <https://id.iste.org/docs/et-documents/computational-thinking-operational-definition-flyer.pdf>

PETRE, M. Why looking isn't always seeing: readership skills and graphical programming. **Communications of the ACM**, v. 38, n. 6, p. 33–44, 1 jun. 1995.

PIAGET, J. **THE ORIGINS OF INTELLIGENCE IN CHILDREN**. 2. ed. NY: International Universities Press Inc., 1956.

REPENNING, A.; BASAWAPATNA, A.; ESCHERLE, N. **Computational thinking tools**. 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). **Anais...** In: 2016 IEEE SYMPOSIUM ON VISUAL LANGUAGES AND HUMAN-CENTRIC COMPUTING (VL/HCC). Cambridge, United Kingdom: IEEE, set. 2016. Disponível em: <http://ieeexplore.ieee.org/document/7739688/>. Acesso em: 1 jun. 2020

RESNICK, M. et al. Scratch: programming for all. **Communications of the ACM**, v. 52, n. 11, p. 60–67, nov. 2009.

SALEN, K.; TEKINBAS, K. S.; ZIMMERMAN, E. **The Game Design Reader: A Rules of Play Anthology**. [s.l.] MIT Press, 2006.

SARIKAS, C. **Vygotsky Scaffolding: What It Is and How to Use It**. Disponível em: <https://blog.prepscholar.com/vygotsky-scaffolding-zone-of-proximal-development>. Acesso em: 18 jun. 2020.

TAUCEI, B. B. Endo-GDC: Desenvolvimento de um Game Design Canvas para Jogos Sérios Endógenos. p. 105, [s.d.].

TCH060: C++ Programming. Disponível em: <https://www.k12.com/high-school-course-list/cplusplus-programming-elective-tch060.html>. Acesso em: 23 jun. 2020.

Thinking concurrently (2.1.5). Disponível em:
<https://www.ocr.org.uk/qualifications/as-a-level-gce-computer-science-h046-h446-from-2015/delivery-guide/component-cs04-02-algorithms-and-programming/delivery-guide-csdg018-thinking-concurrently-215>.

VAHLO, J. et al. Digital Game Dynamics Preferences and Player Types. **Journal of Computer-Mediated Communication**, v. 22, n. 2, p. 88–103, 1 mar. 2017.

VON WANGENHEIM, C. G.; VON WANGENHEIM, A. **Ensinando Computação com Jogos**. Florianópolis: Bookess Editora, 2012. v. 1

WING, J. M. Computational Thinking. **Communications of the ACM**, v. 49, n. 3, p. 33–35, mar. 2006.

WING, J. M. Computational Thinking: What and Why? **Link Magazine**, p. 1–6, 17 nov. 2010.

XEXÉO, G. et al. **UMA INTRODUÇÃO AO OBJETO DE ESTUDO DO LUDES**. [s.l: s.n.].

APÊNDICES

APÊNDICE A – GAME DESIGN CANVAS PARA JOGOS ENDÓGENOS (ENDO-GDC) DE LEO & MAYA

1) PROBLEMA		2) JOGADOR ALUNO	
Oferecer um primeiro contato com PC		Faixa etária: 7 a 12 anos	Todos os gêneros
Oferecer uma introdução ao conceito de PC concorrente		Estudante de ensino fundamental	Alfabetizado
Primeiro contato com PC costuma exigir uma base que pessoas normalmente não possuem		Cenário de uso: no celular num momento de lazer	Cenário de uso: no computador num momento de ócio

3) CONTEÚDO PEDAGÓGICO		7) HISTÓRIA	
Variáveis	Concorrência	Cão e gato bombeiros ajudam animais em apuros pelos quintais	Cada bombeiro tem uma habilidade que os tornam indispensáveis
Condicionais	Barreiras		
Laços	Conceitos básicos de algoritmo		

4) OBJETIVOS DO APRENDIZADO	5) FEEDBACKS EDUCATIVOS	9) ESTÉTICA		8) OBJETIVOS DO JOGO
Reconhecer subproblemas	Mensagem falando o objetivo de cada fase no início	Satisfação ao resolver as fases	Fofura e meiguice	Ir para lugares específicos
Interpretar os objetivos da fase	Mensagem de erro caso com explicação caso o jogador não complete o objetivo	Calma para resolver as fases	Pixel art	Levar ovos para ninhos
analisar as limitações da fase	Botão de tutorial explicando mecânicas do jogo	10) DINÂMICAS		Levar filhotes de passarinho para copa de árvores
conceber algoritmos	Mensagem positiva ao final da fase quando o jogador completar a fase	"Solving problems that require logic, reasoning or analytic thinking"	Carregar objetos de um lugar ao outro	Levar esquilos para o alto de árvores
	Instruções destacadas na trilha conforme execução	Montar algoritmos	Subir e descer de árvores	
		Administrar dois personagens agindo de forma concorrente		

6) INSPIRAÇÕES		11) MECÂNICAS		12) RESTRIÇÕES	
Human Resource Machine	Coding Planets	Trilha de instruções de tamanho limitado (conceber algoritmos)	Definir laços de instruções (conceber algoritmos, analisar limitações da fase, reconhecer subproblemas)	TCC	Unity
Opus Magnum	Lightbot: Code Hour	Instrução de pegar (conceber algoritmos, interpretar objetivo da fase, análise de objetos)	Instrução de andar (conceber algoritmos, interpretar objetivos da fase, analisar objetos)	Computador	Celulares Android
		árvores (conceber algoritmos, analisar limitações da fase)	objetos empurráveis (conceber algoritmos, analisar limitações da fase)	Dois desenvolvedores	Pixel Art
		instrução de barreira (conceber algoritmos, analisar limitações da fase, reconhecer subproblemas)	Mais de um personagem jogável em uma fase (conceber algoritmos, analisar limitações, interpretar objetivo, reconhecer subproblemas)	Jogo digital	Público infantil

APÊNDICE B – DOCUMENTO DE GAME DESIGN (GDD) DO JOGO LEO & MAYA, ORGANIZADO CONFORME DESCRITO POR TAUCEI (2019)

Bloco de Usuário

O Problema

O jogo Leo & Maya existe para tentar sanar, na medida da capacidade, um grande problema percebido pelos alunos envolvidos em sua produção. Essa questão é explicitada no parágrafo abaixo.

O problema é o contato tardio que pessoas costumam ter com pensamento computacional em vida, o que vem a afastar pessoas de campos de exatas. Quando as pessoas têm contato com o modo de pensar computacional, isso normalmente acontece em salas de aula cheias ou no ambiente profissional, sem garantia de que o indivíduo vai conseguir se adaptar ou adquirir as competências necessárias para conceber soluções na forma de instruções que podem ser computáveis. Isso é um problema a partir do momento em que pensamento computacional se torna uma competência cada vez mais importante, conforme cada vez mais computadores se tornam mais presentes na vida cotidiana.

Não só introduzir ao pensamento computacional, o jogo também propõe introduzir seus jogadores a pensar de maneira concorrente. Pensamento computacional concorrente, apesar de ser uma forma menos explorada de se pensar, também tem ganho importância conforme cada vez mais computadores se mostram capazes de rodar código de forma concorrente.

O Nosso Jogador

De modo a ampliar o nosso impacto, nós visamos como principal público-alvo crianças de 7 a 11 anos, sem um gênero específico em mente. Para fins de inclusão, o jogo buscará ao máximo minimizar a necessidade de comunicação verbal, de modo que possa ser jogado por pessoas com variados níveis de escolaridade. Essa faixa etária foi escolhida de acordo com a teoria de desenvolvimento cognitivo de Piaget, que aponta a referida faixa etária como ideal para se ensinar a relacionar símbolos com objetos concretos, traçando desta forma relacionamentos entre instruções e ações tomadas pelos personagens do jogo.

O nosso jogador-alvo possui acesso a um computador com internet ou a um celular Android em momentos de ócio para jogar o jogo. O jogo foi pensado para ser tanto uma atividade solitária, onde o jogador nos próprios termos formula soluções para as fases dos jogos, quanto uma atividade coletiva, onde dois ou mais jogadores discutem soluções para as fases.

Bloco de Aprendizado

Conteúdo Pedagógico

O conteúdo pedagógico foi escolhido com base em currículos lecionados por instituições de ensino ao redor do mundo que visam o ensino de programação. São os conteúdos pedagógicos:

- o que são algoritmos;
- como montar um algoritmo para quando uma entidade solitária for resolver o problema;
- como montar um algoritmo para quando mais de uma entidade vai resolver o problema;

- como trabalhar com variáveis;
- como instruções podem reagir em situações diferentes (if-else);
- como declarar um laço de instruções que se repetirão em *loop*;
- como usar barreiras para controlar fluxos de execução concorrentes.

Objetivos de Aprendizado

Os objetivos de aprendizado do jogo foram decididos com base nos pilares que fundamentam o pensamento computacional e na taxonomia revisada de Bloom. No final do dia, são quatro os objetivos de aprendizado:

- Reconhecimento de subproblemas;
- Interpretação dos objetivos da fase;
- Análise das limitações da fase;
- Concepção de algoritmos.

A ideia é que esses quatro objetivos de aprendizado se relacionem com os pilares de concepção de algoritmos e de decomposição de problemas.

Bloco de Narrativa

História

A história do jogo existe meramente como um conceito. O jogo conta a história de dois animais bombeiros: Léo, um gato novato, e Maya, uma cachorra veterana, que juntos trabalham ajudando animais em apuros em quintais da vizinhança.

Objetivos do Jogo

Os objetivos do jogo visam reforçar a pouca história que se tem. Todos os objetivos giram em torno de montar algoritmos que resolvam as fases, e cada fase apresenta um contexto diferente para o algoritmo.

As fases giram em torno de:

- Colocar esquilos no alto de árvores
- Levar ovos de passarinhos para ninhos
- Tirar filhotes do alto de árvores

Bloco de Gameplay

Mecânicas

As mecânicas estão intimamente ligadas com os objetivos de aprendizado citados acima. São elas:

- Trilha de instruções de tamanho limitado, relacionado com concepção de algoritmos;
- Instrução de andar, relacionado com concepção de algoritmos, análise de limitações da fase e interpretação de fases;
- Instrução de pegar, relacionado com concepção de algoritmos, análise de limitações da fase e interpretação de fases;

- Definir laços de instruções, relacionado com concepção de algoritmos, análise das limitações da fase e reconhecimento de subproblemas;
- Árvores, relacionado com concepção de algoritmos e análise das limitações da fase;
- Objetos empurráveis, relacionado com concepção de algoritmos e análise das limitações da fase;
- Mais de um personagem jogável, relacionado com concepção de algoritmos, análise de limitações da fase, interpretação de objetivos da fase e reconhecimento de subproblemas;
- Instrução de barreira, relacionado com concepção de algoritmos, análise das limitações da fase e reconhecimento de subproblemas;

Dinâmicas

As dinâmicas são:

- Carregar props de um lugar ao outro;
- Montar algoritmos;
- Subir em árvores para alcançar a copa;
- Administrar dois personagens que agem de forma concorrente.

De acordo com “Digital Game Dynamics Preferences and Player Types”, as dinâmicas descritas podem ser resumidas por “Solving problems that require logic, reasoning or analytic thinking”

Estética

A estética que buscamos é promover satisfação após resolver as fases, calma durante o processo de resolver as fases, e fofura e meiguice ao olhar para os personagens. O estilo artístico adotado foi pixel art, pela rapidez.

Bloco de Feedback

A ideia é que o jogador tenha sempre, no menu de fases, um botão de tutorial que mostre como lidar com a mecânica introduzida naquele batch de fases. Desta forma, o jogador sempre pode tirar dúvida sobre o funcionamento de alguma mecânica específica que será necessária não só para o grupo atual de fases, como provavelmente para os futuros.

Também, no início de cada fase, o jogador é tratado com uma mensagem breve explicando o objetivo da fase em questão, e uma mensagem que varia de acordo com o estado final da fase uma vez que o algoritmo do jogador é resolvido. Caso o algoritmo tenha sido suficiente para concluir a fase sem fazer os personagens baterem contra paredes ou objetos intransponíveis, uma mensagem positiva aparece. Caso o algoritmo tenha sido insuficiente, uma mensagem aparece explicando o que deu errado com a resolução. Durante a execução do algoritmo, o jogador vê as ações sendo tomadas conforme as instruções correspondentes na trilha são destacadas.

Bloco de Tecnologias

Tecnologias

As tecnologias utilizadas na concepção do jogo são softwares gratuitos ou devidamente pagos pelos desenvolvedores.

Em sua maior parte, o jogo foi concebido usando a Unity Engine com suporte do Visual Studio Code, duas ferramentas gratuitas. Os sprites foram desenhados e animados no programa pago Aseprite.

Plataformas

O jogo foi feito para duas plataformas: celulares Android e navegadores web capazes de rodar um jogo Unity.

Idealmente, num futuro próximo, podem existir versões para celulares Apple.

Bloco de Inspirações

Nossas maiores inspirações foram jogos similares que estimulem que o jogador programe soluções para problemas.

De um lado, tivemos como inspiração jogos de celular como Lightbot: Code Hour e Coding Planets, como prova de conceito que um jogo desses era possível de se executar em aparelhos móveis e que havia público para tais jogos.

Por outro lado, tivemos como inspirações tanto Human Resource Machine quanto jogos da Zachtronics, como Opus Magnum e TIS-100, que são jogos de criar algoritmos para serem seguidos. Esses jogos serviram de inspiração ao mesclar as próprias narrativas ao ato de jogar, ao oferecer soluções interessantes de usabilidade e de oferecer feedback ao jogador das próprias ações conforme elas acontecem.

APÊNDICE C - Plano de fases do jogo Leo & Maya

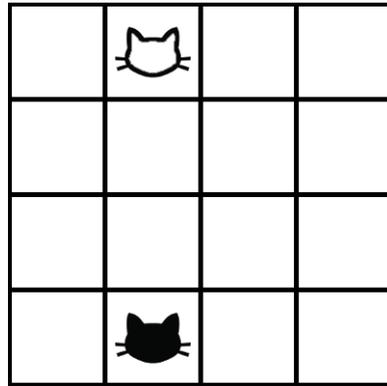
Legenda

	Gato
	Local onde gato deve chegar
	Cachorra
	Local onde cachorra deve chegar
	Obstáculo ou parede
	Objeto que pode ser empurrado
	Animal em apuros
	Local onde se deve entregar o(s) animal(is) em apuros
	Item perdido
	Local onde se deve entregar o item perdido
	Árvore
	Copa da árvore com N passarinhos

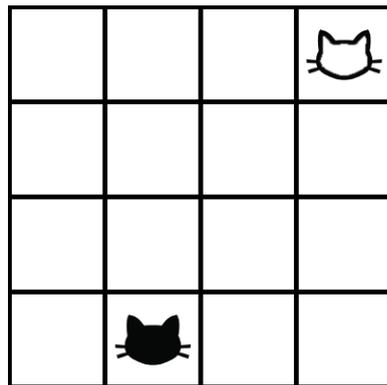
Plano de Fases

1. Andar

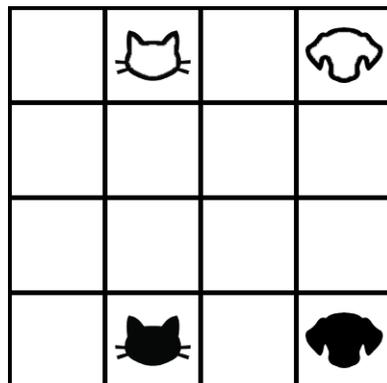
1.1. Aprende a andar



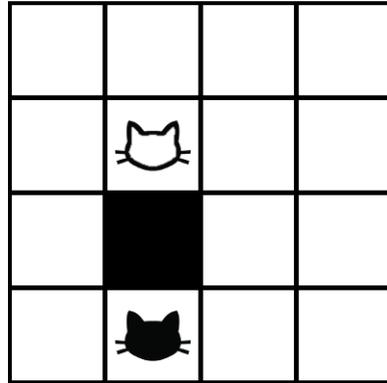
1.2. Aprende a andar em duas direções na mesma fase



1.3. Aprende a andar com dois personagens em paralelo

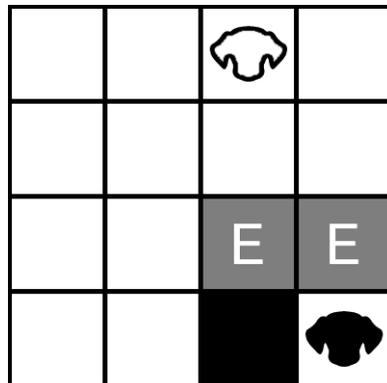


1.4. Aprende a lidar com obstáculos

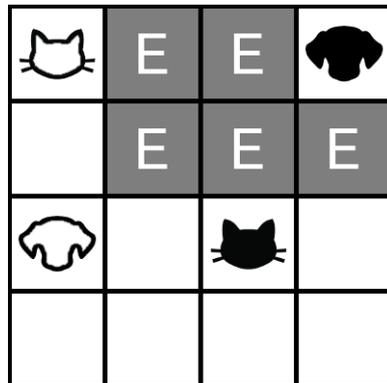


2. Caixas

2.1. Aprende a empurrar caixas

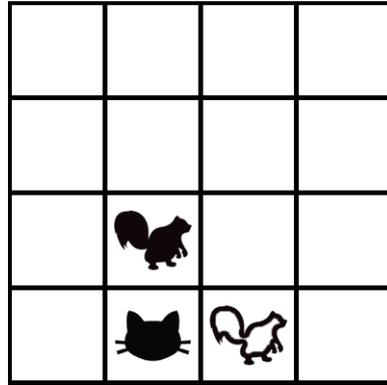


2.2. Aprende a empurrar caixas com paralelismo

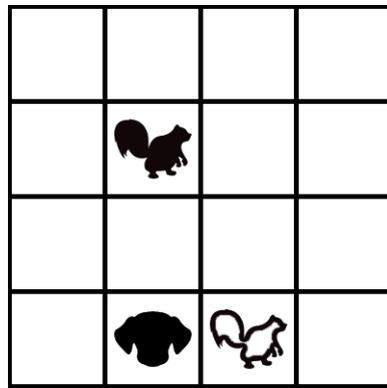


3. Pegar e largar

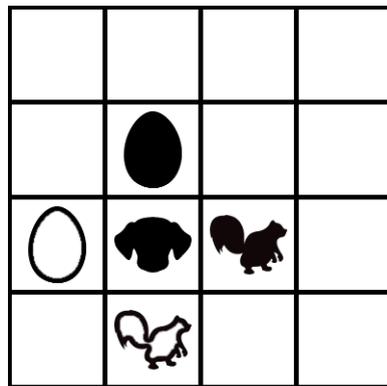
3.1. Aprende a pegar e largar



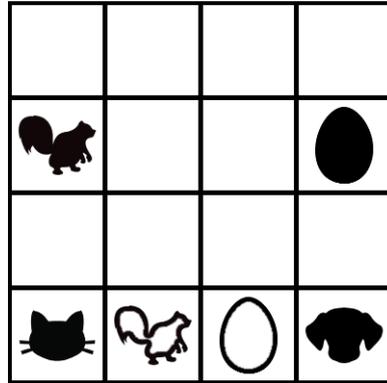
3.2. Aprende a andar, pegar e largar na mesma fase



3.3. Aprende a lidar com mais de um "item pegável" na mesma fase

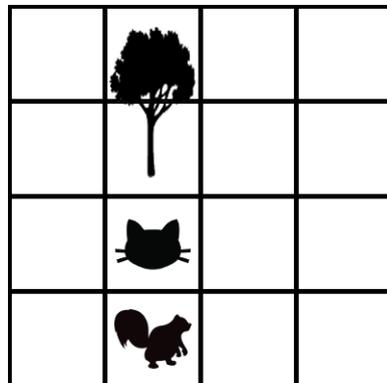


3.4. Aprende a pegar e largar com dois personagens em paralelo



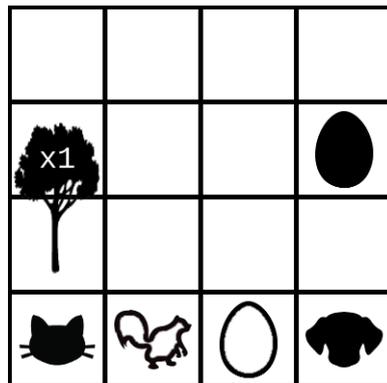
4. Árvores

4.1. Aprende a subir árvore

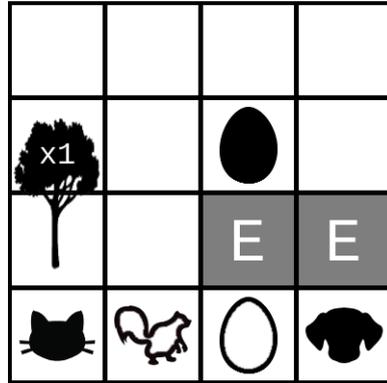


- Objetivo é pegar o animal e colocá-lo na copa da árvore

4.2. Aprende a subir árvore com paralelismo

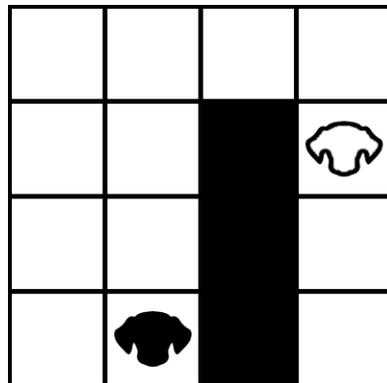


4.3. Aprende a subir árvore e empurrar caixa na mesma fase

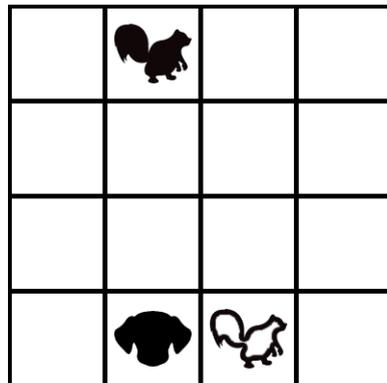


5. Laços

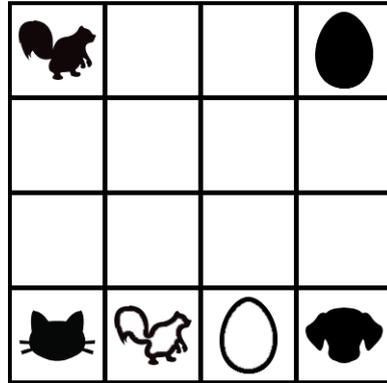
5.1. Aprende a usar laço para repetir instrução



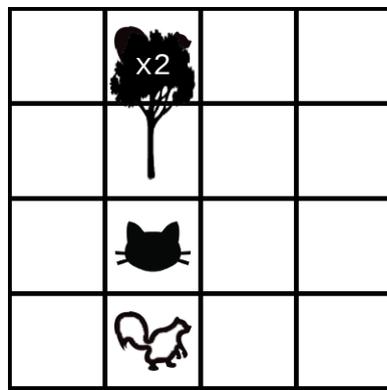
5.2. Aprende a usar mais de um laço na mesma fase



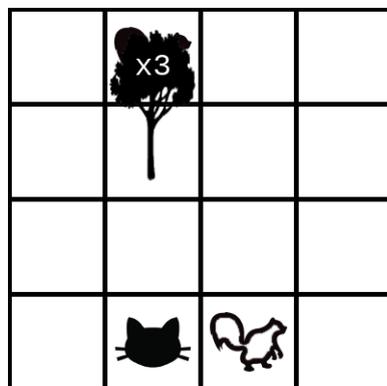
5.3. Aprende a usar laços com paralelismo



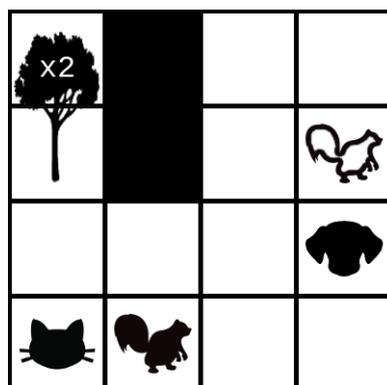
5.4. Aprende a usar laço para repetir conjunto de instruções



5.5. Aprende a usar laço dentro de laço

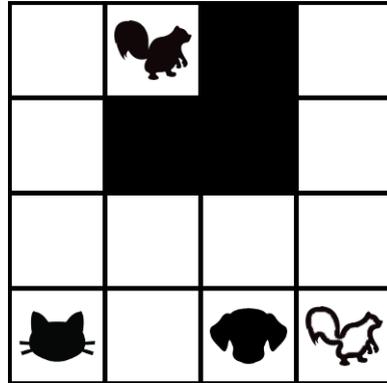


5.6. Aprende a usar laços dentro de laços com paralelismo

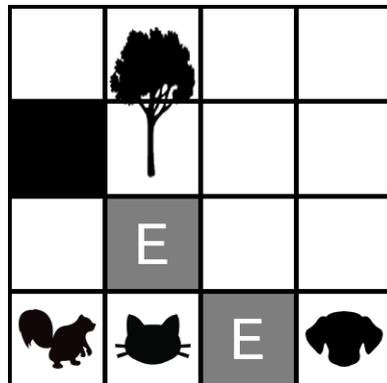


6. Concorrência

6.1. Aprende a usar barreiras de concorrência



6.2. Aprende a usar barreiras, caixas e árvores na mesma fase



- Objetivo é pegar o passarinho e colocá-lo na copa da árvore

6.3. Aprende a usar barreiras dentro de loops

