



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

MagicBits: Um Game Auxiliar para a Matéria de Introdução aos Sistemas Computacionais

Alexandre S. C. Oliveira
Eduardo F. Assis

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Marcus Vinicius Lamar

Brasília
2023

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Marcelo Grandi Mandelli

Banca examinadora composta por:

Prof. Dr. Marcus Vinicius Lamar (Orientador) — UnB/IE/CIC
Prof.^a Dr.^a Carla Denise Castanho — UnB/IE/CIC
Prof. Dr. Ricardo Pezzuol Jacobi — UnB/IE/CIC

CIP — Catalogação Internacional na Publicação

Oliveira, Alexandre S. C..

MagicBits: Um Game Auxiliar para a Matéria de Introdução aos Sistemas Computacionais / Alexandre S. C. Oliveira, Eduardo F. Assis.
Brasília : UnB, 2023.

110 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2023.

1. Jogo educacional, 2. Gamificação, 3. Sistemas Computacionais

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil

Dedicatória

Dedicamos este trabalho a nossas famílias, amigos e a todas as pessoas que contribuíram para este trabalho.

Agradecimentos

Gostaríamos de agradecer ao nosso orientador Prof. Dr. Marcus Vinícius Lamar, pelo grande apoio durante todo o desenvolvimento desse trabalho. Sempre disponível para nos auxiliar, sua dedicação e comentários foram muito relevantes. Agradecemos a todos os alunos da turma de Introdução aos Sistemas Computacionais (ISC) pela participação na pesquisa e seus *feedbacks*. Agradecemos a todos os professores no evento do WPOS que comentaram sobre o trabalho.

Também gostaríamos de agradecer a nossas famílias pelo suporte para que pudéssemos dedicar o tempo e os recursos necessários para o desenvolvimento do estudo.

Agradecemos a todos os nossos amigos que fizeram parte da nossa vida acadêmica, especialmente ao Oscar Etcheverry, Thiago Ferreira, Isabelle Alex, Victor Carneiro, Emanuel Johannes e ao Gabriel Pinheiro.

E por último, mas não menos importante, a todos os autores de imagens, áudios e demais *assets* utilizados no desenvolvimento da pesquisa.

Resumo

Este trabalho¹ apresenta o jogo educacional MagicBits, desenvolvido para oferecer uma alternativa à prática dos conteúdos da disciplina Introdução aos Sistemas Computacionais. O projeto é utilizado como uma ferramenta de apoio ao ensino, e tem o objetivo de aumentar o engajamento dos alunos na disciplina. O MagicBits é constituído de uma série de minigames, os quais oferecem um ambiente lúdico e imersivo, onde os estudantes se tornam protagonistas e devem superar os desafios encontrados. Estes minigames são desenvolvidos para abordar temas específicos da disciplina, oferecendo formas para os alunos aplicarem e reforçarem os conhecimentos adquiridos em sala de aula. O projeto foi aplicado na turma do primeiro semestre de 2023, e pôde-se identificar indícios de uma melhora no desempenho dos alunos em comparação a semestres anteriores.

Palavras-chave: Jogo educacional, Gamificação, Sistemas Computacionais

¹Devido à necessidade de renderização de algumas figuras, os autores orientam o leitor a usar um dos seguintes softwares gratuitos: Adobe Acrobat ReaderTM ou Okular.

Abstract

This work² presents the education game MagicBits, developed to offer an alternative way to students practice the contents of the discipline Introduction to Computing Systems. The project is utilized as a teaching support tool and aims to increase students' engagement in the discipline. MagicBits consists of a series of minigames providing a playful and immersive environment where students become protagonists and must overcome challenges. These minigames are designed to address specific course topics, providing opportunities for students to apply and reinforce the knowledge acquired in the classroom. The experiments were carried out in the first semester of 2023 class, revealing indications of an improvement in student performance compared to previous semesters.

Keywords: Educational Game, Gamification, Computational Systems

²Due to the need to render some figures, the authors guide the reader to use one of the following free software: Adobe Acrobat ReaderTM or Okular.

Sumário

1	Introdução	1
1.1	Problema	2
1.2	Objetivo	4
1.3	Estrutura da Monografia	4
2	Fundamentação Teórica	5
2.1	Jogos sérios e gamificação	5
2.2	Introdução aos Sistemas Computacionais	6
2.3	SCORM	7
2.4	Unity	8
2.5	Trabalhos relacionados	9
3	Metodologia Proposta	11
3.1	Planejamento	12
3.2	Minigames	13
3.2.1	Minigame 1.1	17
3.2.2	Minigame 1.2	25
3.2.3	Minigame 2.1	34
3.2.4	Minigame 2.2	38
3.2.5	Minigame 2.3	47
3.2.6	Minigame 2.4	56
3.3	Detalhes de Implementação	65
4	Resultados Obtidos	69
4.1	Análise das notas com semestres anteriores	69
4.2	Impacto dos minigames nas provas	72
4.3	Correlação entre prova e minigames	74
4.4	Opinião dos alunos	76

5 Conclusão	78
5.1 Trabalhos Futuros	79
Referências	81
Apêndice	84
A Formulário de Avaliação da disciplina ISC - 2022.1	85
B Formulário de Avaliação da disciplina ISC - 2023.1	92

Lista de Figuras

3.1	Tela inicial dos minigames.	14
3.2	UI principal com tempo e quantidade de falhas.	14
3.3	Menu de Pausa.	15
3.4	UI de Créditos.	15
3.5	UI de Conclusão dos minigames	16
3.6	Tutorial inicial.	17
3.7	Cenário inicial do minigame 1.1.	18
3.8	Diálogo entre jogador e Donzela.	18
3.9	Primeiro estágio do desafio sobre álgebra booleana.	19
3.10	Tocha do primeiro estágio do desafio sobre álgebra booleana.	20
3.11	Jogador pega o fogo da tocha para avançar no desafio.	20
3.12	Terceiro estágio do desafio sobre álgebra booleana.	21
3.13	Tocha do terceiro estágio do desafio sobre álgebra booleana.	21
3.14	Diálogo entre jogador e Cavaleiro Trapaceiro.	22
3.15	Cavaleiro Trapaceiro informando o número e a base ao jogador.	23
3.16	Pássaros gerados automaticamente com números na base determinada.	23
3.17	Posição final do Cavaleiro Trapaceiro.	24
3.18	Último desafio do primeiro minigame.	25
3.19	Donzela apresentando o Desafio dos Potes.	26
3.20	Início do Desafio dos Potes.	27
3.21	Jogador preenchendo os potes para o valor especificado pela <i>Donzela</i>	27
3.22	Efeito audiovisual ao jogador acertar a resposta do Desafio dos Potes.	28
3.23	Elevador à direita do jogador para levá-lo ao próximo puzzle.	29
3.24	Cenário antes do Desafio das Prateleiras.	29
3.25	<i>Senhor</i> explicando o que deve ser feito.	30
3.26	Jogador buscando a resposta e retornando ao <i>Senhor</i>	31
3.27	<i>Senhor</i> apresentando terceiro estágio do desafio.	31
3.28	Diálogo entre jogador e Esgrimista.	32
3.29	Interface gráfica com a questão de Complemento de Base.	33

3.30	Interface gráfica com a questão de Ponto Flutuante.	33
3.31	Diálogo entre jogador e Donzela.	34
3.32	Primeiro e mais fácil desafio que o estudante enfrenta no minigame.	35
3.33	Segundo desafio do minigame.	36
3.34	Jogador completando o segundo desafio.	36
3.35	Diálogo com o personagem Fantasma.	37
3.36	Jogador no último desafio e Inimigo patrulhando o caminho.	38
3.37	Início do minigame 2.2.	39
3.38	Diálogo com o Fantasma no Desafio da Ponte.	39
3.39	Jogador no Desafio da Ponte.	40
3.40	Diálogo com o Fantasma no Desafio da Porta.	41
3.41	Desafio da Porta.	42
3.42	Jogador no Desafio dos Lasers.	43
3.43	Desafio dos Fantasmas.	44
3.44	Jogador utilizando o círculo mágico para derrotar o fantasma.	44
3.45	Diálogo com o Viajante no Desafio do Necromante.	45
3.46	Necromante avançando sobre os obstáculos.	46
3.47	Câmera focando o jogador no Domo Protetor.	47
3.48	Cenário inicial do minigame 2.3.	48
3.49	Vale dos Portais no Desafio dos Portais.	48
3.50	Dinâmica de elevação de rochas com proximidade do jogador.	49
3.51	Desafio do Acesso à Memória	50
3.52	Efeitos do Desafio de Acesso a Memória.	51
3.53	Segundo Portal de Saltos do minigame.	51
3.54	UI do desafio sobre Memória de Código.	52
3.55	Cenário próximo ao Desafio da Comida.	53
3.56	Interface do Desafio da Comida.	54
3.57	Cenário do Desafio de Memória de Dados.	55
3.58	Desafio de Memória de Dados.	56
3.59	Donzela introduzindo o minigame.	57
3.60	Desfiladeiro com o primeiro desafio.	57
3.61	UI do puzzle de codificação de instrução.	58
3.62	Jogador inserindo a resposta no <i>AssemblyPuzzle</i>	58
3.63	NPC <i>Fantasma</i> à direita do jogador.	59
3.64	Jogador inserindo resposta no painel.	60
3.65	Ao acertar, o baú se abre e 1 gema é adquirida.	60
3.66	Vilarejo abandonado.	61

3.67	Vilarejo abandonado.	62
3.68	<i>DisassemblyPuzzle</i> do portal.	62
3.69	<i>AssemblyPuzzle</i> das lanças.	63
3.70	<i>AssemblyPuzzle</i> pedindo codificação em hexadecimal.	64
3.71	<i>AssemblyPuzzle</i> pedindo codificação em hexadecimal.	65
3.72	Vila restaurada e tela de conclusão.	65
3.73	Diagrama de relações entre os componentes do Sistema de Diálogo.	67
3.74	Exemplo de diálogo aberto no <i>TalkerMaker Deluxe</i>	68
4.1	Histogramas da primeira prova.	70
4.2	Histogramas da segunda prova.	71
4.3	Relação entre notas do primeiro módulo.	75
4.4	Relação entre notas do segundo módulo.	76

Lista de Tabelas

4.1	Resultados das provas dos semestres entre 2023.1 e 2021.2.	71
4.2	Quantidade de alunos que abriram ou não abriram os minigames	72
4.3	Quantidade de alunos com bom desempenho.	73
4.4	Quantidade de alunos com baixo desempenho.	74

Lista de Abreviaturas e Siglas

ISC Introdução aos Sistemas Computacionais.

LMS Sistema de Gestão da Aprendizagem (do inglês, *Learning Management System*).

Moodle Ambiente Modular de Aprendizagem Dinâmica Orientada a Objetos (do inglês, *Modular Object-Oriented Dynamic Learning Environment*).

NPC Personagem não jogável (do inglês, *Non Playable Character*).

RPG jogo de interpretação de papéis (do inglês, *Role-playing game*).

SCORM Modelo Referência de Objeto para Compartilhamento de Dados (do inglês, *Sharable Content Object Reference Model*).

TIC Tecnologia de Informação e Comunicação.

UI Interface de Usuário (do inglês, *User Interface*).

UnB Universidade de Brasília.

VFX Efeitos Visuais (do inglês, *Visual Effects*).

Capítulo 1

Introdução

Atualmente, o uso da tecnologia tem contribuído significativamente para melhorar o processo de ensino-aprendizagem. Dentre as estratégias utilizadas, a gamificação é uma das mais adotadas para aumentar o engajamento dos alunos em relação aos temas abordados nas disciplinas [1].

Os “nativos digitais” [2], como são chamadas as novas gerações, são pessoas que tiveram acesso à tecnologias do século XXI, como computadores, *smartphones*, *tablets*, videogames, desde muito jovens. Dessa forma, esses indivíduos são proficientes em manipular esses aparelhos e o fazem de forma natural. Conforme os autores de [1], “As novas gerações não se satisfazem em ler manuais técnicos ou instruções, preferem “aprender fazendo”, pois já o fazem naturalmente quando, por exemplo, descobrem como funciona um novo dispositivo ou um novo jogo de videogame”.

A forma de ensino tradicional, onde professor é responsável por transmitir novos conceitos aos estudantes, e os materiais didáticos consistindo em livros em sua grande maioria, se mostra um tanto abstrata e falha em manter níveis de atenção e motivação necessários para o aprendizado ocorrer. Assim, é observado que metodologias de ensino mais práticas e alinhadas às características das novas gerações se mostra necessária.

Nesse contexto, a gamificação é vista como uma ótima alternativa à forma tradicional de ensino. Através da aplicação de elementos de jogos, e um processo de *design* cuidadoso, diversas pesquisas mostram que essa metodologia promove o aumento da motivação e engajamento dos alunos, e, conseqüentemente, uma melhora na aprendizagem [3].

Através da construção de um conjunto de jogos educativos, o estudo vigente visa gamificar a matéria de Introdução aos Sistemas Computacionais (ISC) da Universidade de Brasília (UnB), uma matéria introdutória do curso de “Ciência da Computação”. Através de métricas colhidas durante os experimentos e notas de avaliações da disciplina, é analisado se a utilização de elementos de jogos têm um impacto positivo no desempenho dos alunos.

1.1 Problema

Geralmente nos cursos de Bacharelado em Ciência da Computação, Engenharia da Computação e afins, existe no currículo uma disciplina de primeiro semestre que introduz os conceitos básicos dos sistemas computacionais modernos. Os conteúdos desta disciplina variam de universidade para universidade, mas geralmente abordam temas como a história da computação, sistemas numéricos, representação de dados, uma introdução sobre como os processadores funcionam e a linguagem Assembly.

A inclusão de uma disciplina como Introdução aos Sistemas Computacionais (ISC) no currículo tem também como objetivo fornecer aos alunos uma visão geral do curso que escolheram, a fim de reduzir a taxa de evasão do curso nos semestres seguintes. As reprovações em geral ocorrem devido ao abandono da disciplina (ou do curso), mas também são identificados alunos com dificuldades na compreensão do conteúdo.

A disciplina ISC é um desafio significativo para muitos alunos, especialmente aqueles sem experiência prévia em computação, devido à sua posição inicial no currículo do curso. A disparidade na formação e nos conhecimentos prévios é uma das principais causas dessas dificuldades. Alunos provenientes de escolas públicas muitas vezes não tiveram a oportunidade de estudar certos temas ou, quando o fizeram, receberam apenas uma introdução muito superficial.

Com o objetivo de identificar com maior precisão as dificuldades dos alunos e encontrar pontos de melhorias, ao final do semestre 2022.1 (calendário da UnB), na primeira semana do mês de outubro, foi aplicado um questionário anônimo aos alunos de ISC com intuito de coletar *feedbacks* sobre suas percepções em relação à disciplina, abordando tópicos como as dificuldades enfrentadas, do que gostaram e sugestões de melhorias para a disciplina, conforme o Anexo A.

Dos 46 alunos inicialmente matriculados na turma, 18 responderam ao questionário. Das perguntas que podiam ser respondidas em uma escala de 1 a 5, em que 1 representa pouco e 5 representa muito, os resultados foram os seguintes:

- “O quão difícil você achou a disciplina ISC?”, recebeu uma média de 4.06;
- “O quão bem você acha que conseguiu absorver o conteúdo de ISC?”, recebeu uma média de 3.56.

Quanto à pergunta discursiva “Qual conteúdo de ISC você achou mais difícil? Por quê?”, 28% das respostas incluíam a palavra “*Assembly*”. Algumas das respostas para essa pergunta foram:

- “*Assembly*. Meu primeiro contato com linguagem de programação foi Python, uma linguagem de alto nível. Já *Assembly* é uma linguagem de baixo nível, logo tive uma surpresa com a matéria”;
- “Elétrica, porque eu não havia estudado nada na escola. E montagem e desmontagem de programas em *assembly* (não programar em *assembly*, mas converter/desconverter para binário), porque eu me perco”;
- “Sistemas Numéricos. Não sei porque. Não me dei bem com esse conteúdo, na minha cabeça essa lógica é super confusa e com muitos detalhes pequenos para recordar”;
- “Conversão de bases, pois foi um período de adaptação minha à universidade e acho que houveram alguns “pulos” na explicação da matéria”.

Na pergunta “Como você acha que a matéria ISC poderia ser melhorada? (Justifique seus pontos.)”, 39% das respostas mencionaram a palavra “conteúdo”. Algumas das respostas foram:

- “Poderia melhorar na questão que envolve a abordagem de *Assembly*. Nas aulas são mostradas ideias muito superficiais sobre o assunto, o que se torna um fator de grande impacto no momento da realização do trabalho”;
- “Mais exercícios feitos em aula para praticar o conhecimento. É muito comum de um conteúdo parecer fácil na aula teórica, mas na hora da prova o aluno não saber por onde começar para resolver algum problema relacionado a ele. Passar exercícios e cobrá-los poderá treinar melhor o aluno para a prova, já que muitos não praticam nada até ela. Os testes não são muito eficientes, porque o tempo limitado para resolvê-los (só 7 minutos) desincentiva que o aluno pense direito nas suas respostas e incentiva que ele chute mais ou simplesmente só cole com os colegas”;
- “Poderia ser melhorada com a resolução de exercícios e uma pequena prática”.

Portanto, podemos afirmar que parte dos alunos têm dificuldade em acompanhar as aulas de ISC e acabam não absorvendo o conteúdo de maneira satisfatória ou, até mesmo, acabam reprovando na disciplina. Muitas vezes não por falta de esforço ou descaso, mas sim pela diferença de bagagem que cada um deles chegam na disciplina ou pela dificuldade em compreender os conteúdos na forma que eles estão sendo apresentados.

1.2 Objetivo

O objetivo principal deste trabalho é estudar uma forma de usar gamificação como uma ferramenta de apoio, a fim de aumentar o aproveitamento do aluno. As dinâmicas propostas devem contribuir em manter o aluno engajado. Como consequência, é desejado que os alunos tenham um melhor desempenho na disciplina.

Para alcançar esse objetivo, é proposto a criação de um jogo educacional, chamado *MagicBits*, consistindo em vários minigames desafiadores, onde os alunos precisam aplicar os conhecimentos adquiridos em sala de aula para progredir. Ao concluir cada minigame, o aluno recebe uma nota com base em seu desempenho.

1.3 Estrutura da Monografia

A presente monografia segue a seguinte estrutura. O Capítulo 1 - Introdução apresenta uma exposição inicial do tema, identificação do problema e a estrutura geral do texto. Em seguida, o Capítulo 2 - Fundamentação Teórica explica os conceitos relevantes para o desenvolvimento do texto, bem como a revisão de estudos relacionados. A metodologia proposta é abordada no Capítulo 3, descrevendo o planejamento realizado, o cronograma adotado, a estrutura da aplicação e dos sistemas desenvolvidos, além da monitoração da turma e coleta de resultados. Em seguida, no Capítulo 4 os resultados obtidos são expostos, apresentando os dados coletados durante as experimentações com as turmas de ISC. Por fim, o Capítulo 5 - Conclusão encerra a monografia, trazendo considerações finais sobre o trabalho realizado e apontando possíveis direções para futuras pesquisas.

Capítulo 2

Fundamentação Teórica

Esta monografia apresenta o desenvolvimento e análise do jogo educacional MagicBits. Para isso, precisamos primeiro definir o que é um jogo educacional e entender sua relação com a gamificação.

2.1 Jogos sérios e gamificação

Um jogo pode ser definido como um conjunto de elementos que contribuem para a diversão (*playfulness*) onde se tem um objetivo a ser alcançado [4]. Dando um exemplo, quando uma criança está brincando com seu brinquedo favorito sem um objetivo claro, ela o faz somente pela diversão em si, sem um propósito claro, isso seria o que a literatura chama de *playfulness*. Agora, digamos que seu irmão mais velho a chame para “brincar” de amarelinha, uma dinâmica com regras definidas em que para ganhar é necessário alcançar um objetivo, isso seria um jogo, também referido pela literatura como *gamefulness*. No momento em que a brincadeira passa a ter regras e um marco a ser alcançado o agente se torna um jogador, essas qualidades definem a experiência como um jogo.

Seguindo essa ideia, um jogo educativo é definido pela literatura como um jogo cujo o objetivo principal é ensinar, transmitir algum valor definido ou conhecimento e o meio para esse fim é a experiência divertida e engajante. Em outras palavras, ao mesmo tempo em que o jogo educativo oferece uma experiência rica, com elementos lúdicos, uma narrativa, objetivos e regras, e um contexto delimitado, o seu objetivo final é transmitir conhecimento para o jogador [5]. O jogo educacional é considerado uma vertente do gênero jogo sério [5], outras categorias seriam treinamento, bem-estar, assistência médica [6], entre outros.

Tanto os jogos sérios quanto a gamificação são áreas de estudos recentes que se assemelham em vários aspectos, e mesmo hoje, não existe uma separação clara para quais experiências se enquadram como gamificação e quais como jogos educacionais, sendo um processo na maioria das vezes subjetivo feito pelos autores [4]. No entanto, uma definição

recente para gamificação que vem se consolidando é a de “aplicar elementos de jogos em outro contexto” [4]. Por exemplo, plataformas de apoio ao ensino como o Moodle implementam pontuações, rankings, emblemas e medalhas que são atribuídas ao estudante [7], isso são elementos de jogos e formam uma experiência gamificada.

A gamificação tem se tornado cada vez mais popular nos últimos anos, com um aumento significativo do interesse por experiências interativas desde aproximadamente 2010 [8]. Nesse contexto, os educadores têm buscado modernizar suas aulas e materiais didáticos, incorporando elementos de jogos, para aumentar o engajamento dos alunos [1].

De acordo com o estudo intitulado “Uma Revisão Sistemática do Uso da Gamificação no Ensino de Programação” [9], a aplicação da gamificação em disciplinas de programação tem se mostrado promissora, pois aumenta a motivação dos alunos e, conseqüentemente, melhora o processo de aprendizado. Os autores chegaram a essa conclusão após analisar 23 pesquisas realizadas entre os anos de 2012 e 2020.

A seguir será apresentado a estrutura geral de ISC e os temas abordados pela disciplina. Nas seções subsequentes, serão discutidos os principais conceitos que compõem esta pesquisa.

2.2 Introdução aos Sistemas Computacionais

A disciplina ISC possui carga horária de 60h, é ministrada no primeiro semestre do curso de Bacharelado em Ciência da Computação e é dividida em 3 módulos:

- **1º módulo - Representação de Dados:** Neste módulo são apresentados os conceitos fundamentais dos sistemas computacionais, os sistemas numéricos e a representação de dados, incluindo números (conjuntos \mathbb{N} , \mathbb{Z} e \mathbb{R}), caracteres, sons, imagens e vídeos. Além disso, é discutido o histórico da computação, fornecendo uma visão geral do seu desenvolvimento ao longo do tempo. Também é feita uma introdução à Álgebra Booleana. Por fim, são abordadas as limitações da aritmética computacional, destacando a questão da precisão finita nos cálculos realizados pelos computadores.
- **2º módulo - Processamento dos Dados:** Neste módulo é descrito como o processador realiza o processamento dos dados, abordando o nível mais baixo do processamento. Na disciplina ISC, é feita uma introdução à arquitetura RISC-V (RV32IM). Durante o estudo, são apresentados os diferentes tipos de instruções em Assembly do processador RISC-V de 32 bits, incluindo instruções aritméticas, lógicas, de saltos condicionais e incondicionais e de acessos a dados da memória.

- **3º módulo - O Processador:** Neste módulo os alunos exploram em maior profundidade os aspectos de hardware. Eles adquirem a compreensão de como uma Unidade Central de Processamento (CPU) é construída e das diferentes unidades que a compõem. Além disso, estudam os princípios da eletrônica, desde uma revisão dos conceitos aprendidos no ensino médio, tais como tensão e corrente, passando pelos dispositivos semicondutores (diodo e transistor), até a exploração de elementos mais complexos, como registradores, Unidade Lógica e Aritmética (ULA), Unidade de Controle e Memória.

Ao passar por esses tópicos da computação, o aluno conclui a disciplina com uma visão geral sobre os sistemas computacionais, tanto do ponto de vista de *software* quanto de *hardware*, adquirindo uma base de conhecimento fundamental que será explorada em matérias posteriores do curso. Quanto melhor for a absorção dos conteúdos apresentados em ISC, maiores as chances do aluno de suceder nas matérias mais avançadas.

Visando facilitar o acesso das atividades do MagicBits pelo aluno, e manter o fluxo de navegação para a realização de atividades da disciplina, foi definido que os minigames seriam disponibilizados no Ambiente Modular de Aprendizagem Dinâmica Orientada a Objetos (do inglês, *Modular Object-Oriented Dynamic Learning Environment*) (Moodle), plataforma Sistema de Gestão da Aprendizagem (do inglês, *Learning Management System*) (LMS) utilizada oficialmente pela UnB. Assim, o estudante consegue acessar as novas atividades propostas através dos mesmos passos que já utiliza para acessar qualquer outra atividade da disciplina. Dessa forma, não há necessidade de aprender uma nova forma de acesso ou plataforma adicional, proporcionando uma experiência mais conveniente e intuitiva para o aluno.

Para viabilizar essa integração, foi necessário o uso da tecnologia SCORM. Através dela foi possível hospedar os minigames no Moodle de forma transparente, acessível como qualquer outra atividade pelo aluno.

2.3 SCORM

O Modelo Referência de Objeto para Compartilhamento de Dados (do inglês, *Sharable Content Object Reference Model*) (SCORM) [10] consiste em um conjunto de especificações, chamados de *books*, que padronizam o fornecimento de material didático em plataformas LMS, como o Moodle. O SCORM promove a interoperabilidade, acessibilidade e reusabilidade dos conteúdos *e-learning*, nome genérico dado ao processo de aprendizagem apoiado em Tecnologia de Informação e Comunicação (TIC) [11]. Essas especificações não se referem a implementação ou padrões de projeto, mas sim definem a estrutura de arquivos base do pacote, bem como as interfaces e métodos a serem implementados. O

pacote, por sua vez, é um arquivo compactado. Dessa forma, qualquer material que siga as especificações do SCORM pode ser disponibilizado em plataformas que tenham suporte a ele.

Existem outros padrões com propósito semelhante ao do SCORM, como o *xAPI* [12] e *IMS Common Cartridge* [13]. No entanto, o SCORM é um dos padrões mais antigos e maduros e suas especificações ficam em primeira posição no quesito popularidade [12]. Além disso, pela sua alta taxa de adoção e tempo de existência (a versão 1.0 surgiu em 2000) há uma maior quantidade de casos de uso e documentação sobre como utilizá-lo.

Uma das versões mais populares do SCORM é a versão **1.2**, que foi lançada há mais de 20 anos, em outubro de 2001. Apesar de sua antiguidade, essa versão ainda é amplamente utilizada pelas plataformas de LMS [12]. O Moodle, por exemplo, oferece suporte total à versão **1.2** do SCORM, permitindo a integração de conteúdos desenvolvidos nesse formato.

Além da versão **1.2**, o Moodle também oferece suporte parcial à versão **2004** do SCORM. No entanto, é importante destacar que algumas funcionalidades dessa versão, como a parte de navegação e sequenciamento, não estão totalmente implementadas no Moodle [14].

Considerando esses aspectos, optou-se pela utilização do SCORM na versão **1.2** como formato para disponibilizar as atividades desenvolvidas no Moodle. A escolha dessa versão se deve ao fato de ser uma versão estabelecida e amplamente suportada pela comunidade. Essa decisão permite que as atividades do projeto sejam acessíveis não apenas no Moodle, mas também em diversas outras plataformas de LMS, como o *BlackBoard* [15] e *Canvas* [16], que também oferecem suporte ao SCORM **1.2**.

2.4 Unity

Para desenvolver os sistemas e mecânicas dos minigames, bem como lidar com os recursos visuais e sonoros, foi escolhida a popular *game engine*¹Unity [17]. Essa *engine* é amplamente reconhecida na indústria, juntamente com outras como a *Unreal Engine* e a *Godot* [18]. No contexto específico desta pesquisa, a escolha da Unity se destaca, pois ela é especialmente adequada para o desenvolvimento de jogos *mobile*, ou seja, jogos projetados para serem executados em dispositivos com recursos limitados, como *smartphones* e navegadores web, nos quais o desempenho pode ser inferior devido a restrições de hardware ou ambientes de execução específicos.

¹Uma *Game Engine* (em tradução livre, *Motor de Jogo*) consiste em um software que fornece diversos sistemas e camadas de abstrações para simplificar e acelerar o desenvolvimento de jogos.

Além disso, geralmente não é eficiente desenvolver suas próprias ferramentas e bibliotecas, o tempo investido será muito maior do que o retorno esperado, salvo casos específicos. Dessa forma, bibliotecas como *Three.js* [19] ou *Pixi.js* [20] não foram consideradas, uma vez que uma *engine* “completa”, onde mais sistemas são fornecidos, oferece abstrações e permite aos desenvolvedores direcionarem sua atenção no desenvolvimento da aplicação.

É importante destacar que as APIs dos navegadores são mais limitadas em termos de funcionalidades quando comparadas às APIs de aplicativos nativos de *desktop*. Além disso, os próprios navegadores consomem recursos computacionais adicionais para o seu próprio funcionamento. Como resultado, uma aplicação executada no navegador será mais restrita em termos de funcionalidade e terá um desempenho inferior em comparação a uma aplicação equivalente que seja executada de forma nativa, utilizando as APIs do sistema operacional.

Além da popularidade mencionada anteriormente, entre as três *engines* consideradas, apenas a Unity oferecia um *plugin* em sua loja de *assets* para adicionar suporte ao pacote SCORM². Esse fato foi determinante para a escolha da *engine* no desenvolvimento do projeto e demonstra a influência significativa que a *Asset Store* da Unity possui em relação às outras plataformas.

A linguagem de programação utilizada na Unity é a C#, criada pela Microsoft [21], e é a linguagem onde os sistemas do MagicBits foram desenvolvidos. Essa linguagem faz parte da família de linguagens C e se assemelha bastante ao Java, outra linguagem de programação: ambas apresentam um estilo imperativo de programação, ambas empregam Orientação à Objetos e ambas são interpretadas/híbridas.

2.5 Trabalhos relacionados

No estudo *Uma análise dos tipos de jogadores em uma plataforma de gamificação incorporada a um sistema juiz on-line* [22] os autores identificam os tipos de jogadores na plataforma de gamificação *CodePlay* desenvolvida como um plugin para o sistema juiz on-line [23] *Codebench*. Essa identificação é feita seguindo a taxonomia Hexad. Através desse framework, os alunos são classificados em seis perfis: Filantropo, Socializador, Espírito Livre, Conquistador, Jogador e Disruptor. A plataforma *CodePlay* consiste em um jogo de interpretação de papéis (do inglês, *Role-playing game*) (RPG) multijogador (do inglês, *multiplayer*) que adiciona elementos de jogos no sistema *Codebench*. O objetivo é identificar e analisar as preferências, comportamentos e interesses de cada tipo de jogador, trazendo *insights* importantes para o desenvolvimento e aprimoramento de ambientes gamificados personalizados.

²Informação referente a agosto de 2022.

Os autores de *Project Éden: platform for introductory programming concepts* [24] apresentam uma versão mais completa e divertida do trabalho deles, intitulado “Project Éden”. O projeto consiste em um jogo educacional para melhorar o processo de aprendizagem de programação. Nele, o jogador precisa instanciar variáveis, modificar valores, trabalhar com laços de repetição e fazer a análise de códigos para progredir e alcançar o desafio final. Alunos e professores que participaram do estudo julgaram a experiência como sendo divertida, onde o conteúdo foi abordado de maneira dinâmica e compreensível mesmo considerando pessoas com diferentes graus de escolaridade.

Outro estudo envolvendo disciplinas de programação é o *Maze Code: Retórica Procedural Aplicada ao Ensino de Lógica de Programação* [25]. Nele é desenvolvido o jogo educacional “Maze Code”, que tem como foco o ensino de conceitos envolvendo programação. Isso é feito através de exercícios onde o jogador utiliza blocos para construir a lógica desejada. O protótipo foi avaliado por alunos do curso de Ciência da Computação da Universidade Federal de Viçosa - Campus Florestal e os resultados obtidos reforçam que a retórica procedural é eficaz para o ensino de programação básica no contexto do trabalho. Ao contrário do estudo vigente, os autores do “Maze Code” não utilizaram o desempenho dos alunos na disciplina como critério para análise da efetividade do projeto.

Além da educação, os elementos de jogos também podem ser aplicados em outras áreas, no estudo *A Serious Game Designed to Promote Safe Behaviors Among Health Care Workers During the COVID-19 Pandemic-Development of Escape COVID-19* [6] é desenvolvido um jogo sério para assistência médica (do inglês, *health care*). Nesse estudo, os autores propõe a aplicação de um jogo sério disponibilizado como um pacote SCORM para treinar profissionais da área da medicina sobre os cuidados necessários a respeito do vírus COVID-19.

Para o compreender o cenário atual da gamificação e dos jogos sérios na academia, o artigo *Gamifying education: what is known, what is believed and what remains uncertain: a critical review* [8] é comumente utilizado. Nele é realizado uma revisão exaustiva da literatura de estudos envolvendo a aplicação da gamificação, onde é examinado tanto os resultados reportados quanto a forma como eles foram obtidos. Na conclusão, os autores indicam que existe uma necessidade em mapear, de forma sistemática, os elementos de jogos com a aprendizagem e fatores de motivações de indivíduos ou grupo de indivíduos.

Além disso, o estudo defende a ideia de que gamificação não é somente uma tecnologia e sim uma metodologia, um design de comportamento que pode ser aplicado em diversas áreas. E que para ser efetiva, é crucial entender o público-alvo do sistema gamificado. Especificamente, a escolha dos elementos de jogos a serem empregados deve ser feita baseada nas preferências e necessidades únicas de cada grupo de estudantes.

Capítulo 3

Metodologia Proposta

Este trabalho tem como objetivo desenvolver um conjunto de minigames curtos e dinâmicos, nos quais o aluno se torna um jogador ativo em um universo criado. Através de quebra-cabeças (do Inglês, *puzzles*) e desafios de lógica, as atividades apresentam os conteúdos da disciplina de forma amigável e descontraída. A ideia é oferecer desafios com dificuldade progressiva, não sendo muito fáceis a ponto de serem triviais, nem muito difíceis a ponto de serem frustrantes para o aluno [3]. O ambiente deve ser envolvente o suficiente para manter o aluno engajado para que o mesmo supere os desafios e progrida no jogo.

Cada minigame é responsável por apresentar tópicos específicos da disciplina e é apresentado como uma atividade totalmente integrada à plataforma Moodle. Isso facilita o acesso dos alunos aos recursos. Assim como outras tarefas, o professor tem o controle sobre essa atividade, podendo determinar se ela é obrigatória, se conta como avaliação, se tem um prazo para conclusão, entre outros aspectos.

Os minigames são disponibilizados para alunos de ISC e o desempenho é monitorado através de métricas fornecidas pelo Moodle, bem como por formulários anônimos. Os minigames são agrupados em dois módulos, que correspondem à divisão de conteúdos das provas do 1º módulo e do 2º módulo. Os alunos têm até o dia de cada prova para completar o módulo de minigames correspondente. Essa restrição é importante para a avaliação do impacto dos minigames no desempenho dos alunos nas provas.

Após a aplicação de cada prova, são colhidas as notas das provas e métricas das atividades no Moodle. Em seguida, é realizada uma análise dos resultados obtidos para determinar se os minigames realmente contribuíram para o aprendizado dos alunos ou não, comparando com semestres anteriores quando os minigames não foram utilizados.

Afim de aperfeiçoar a experiência ...

3.1 Planejamento

Os minigames são divididos de forma semelhante ao plano de ensino da disciplina ISC, seguindo a mesma cronologia e disposição de assuntos. Cada jogo é chamado de “minigame” para transmitir a ideia de jogos curtos e rápidos, o aluno gasta em média 20 minutos para completar um minigame. Cada minigame apresenta um conjunto de desafios, também chamados de *puzzles*, onde o aluno tem de aplicar certos conhecimentos vistos em sala de aula para progredir.

É importante ressaltar que o papel do professor em sala de aula não será excluído ou diminuído em importância. Os minigames não apresentarão novos conhecimentos, mas sim reforçarão o que foi ensinado durante as aulas. Em outras palavras, o professor ensinará em sala de aula, apresentando novos conceitos e incentivando a reflexão, enquanto os alunos terão os minigames a sua disposição para exercitar os conceitos aprendidos.

Os minigames são inspirados nos estilos **plataforma** [26], **de rolagem lateral** [27] e de **ação e aventura** [28]. No estilo plataforma, o jogador controla um personagem que explora um mundo fictício, realizando ações como andar, pular e interagir com o ambiente. Já no estilo de rolagem lateral, a câmera segue a visão lateral do personagem e a tela rola horizontalmente a medida que o personagem se move. No estilo de ação e aventura, o jogador assume o papel de um personagem e pode realizar uma variedade de ações, como usar magias, realizar ataques físicos e interagir com outros personagens e objetos.

No desenvolvimento deste trabalho, são utilizados diferentes tipos de recursos, conhecidos como *assets* na indústria de jogos [29]. Esses *assets* podem ser gratuitos ou pagos e englobam desenhos de personagens, cenários, faixas de áudio, efeitos visuais VFX e alguns *scripts*. Dessa forma, a maior parte do trabalho consiste em integrar esses conteúdos ao ambiente criado e desenvolver os sistemas e mecânicas essenciais relacionados aos conteúdos da disciplina de ISC.

Para o semestre de 2022.2¹, foi disponibilizado os minigames do 2º módulo da disciplina. Tal módulo foi priorizado pela necessidade de tempo de desenvolvimento e do plano de aula da matéria ISC. O desenvolvimento do projeto começou no final de agosto de 2022 (i.e., semestre 2022.1 da UnB). Foi previsto um período de 3 meses de desenvolvimento para finalizar uma versão pré-eliminar do MagicBits, composto por diversos minigames. Dessa forma, os minigames referentes ao 2º módulo foram priorizados para que pudessem ser utilizados pela turma do semestre 2022.2.

Assim, foi realizado um teste piloto na turma de 2022.2 para refinar o *game design* [30] do minigames construídos. Ao final do semestre 2022.2, após concluído o piloto, foram coletados comentários a respeito dos minigames aplicados através de um formulário anô-

¹Na UnB, o ano letivo é dividido em dois semestres e representados na forma <ANO>.<SEMESTRE>. Por exemplo, o semestre 2023.1 corresponde ao primeiro semestre do ano de 2023.

nimo. Com essas informações, foi possível melhorar as mecânicas e sistemas criados nas próximas iterações de desenvolvimento do MagicBits, e aprimorar a experiência do jogador.

No início do semestre 2023.1, o desenvolvimento dos minigames dos módulos 1 e 2 foi concluído. Após a aplicação dos minigames, foram conduzidas pesquisas com os alunos para coletar suas opiniões e realizar comparações. Além disso, após a conclusão das provas dos módulos 1 e 2, para os quais os minigames foram desenvolvidos, as notas de cada aluno foram coletadas. Com base nesses dados, uma análise foi feita para avaliar o desempenho da turma em relação aos três semestres anteriores, essa análise é apresentada no Capítulo 4.

3.2 Minigames

Os minigames são jogos curtos, com duração de aproximadamente 20 minutos, nos quais os alunos se tornam protagonistas de um universo fictício e precisam aplicar seus conhecimentos da disciplina para superar desafios. Esses jogos foram criados com o objetivo de proporcionar aos alunos uma experiência prática e interativa, permitindo que eles apliquem o conhecimento adquirido em sala de aula de maneira divertida e desafiadora [1]. Os minigames são divididos em módulos que seguem a sequência e a estrutura dos conteúdos da disciplina. Até o momento da escrita deste texto, foram desenvolvidos dois módulos.

Para o primeiro módulo, foram desenvolvidos dois minigames. O primeiro minigame aborda temas como Álgebra Booleana, conversão de bases e operações simples entre bases distintas. O segundo minigame concentra-se em conversão de bases, introduzindo conceitos como complemento de bases, ponto fixo e ponto flutuante.

Para o segundo módulo, foram desenvolvidos quatro minigames, que exploram o conteúdo relacionado ao *Assembly RISC-V*. O primeiro minigame concentra-se em instruções de operações aritméticas simples. O segundo minigame aborda instruções de deslocamento de bits e lógica booleana. O terceiro minigame explora saltos incondicionais e condicionais, e instruções de acesso à memória de dados e análise de código. Por fim, o quarto minigame aborda a representação em linguagem de máquina.

Os minigames foram nomeados seguindo um padrão de nomenclatura para facilitar a identificação. Cada minigame é nomeado como “minigame x.y”, em que “x” representa o número do módulo e “y” indica a numeração específica do minigame dentro desse módulo.

Todos os minigames possuem uma tela de abertura com uma música e um efeito visual, como pode ser visto na Figura 3.1.

Figura 3.1: Tela inicial dos minigames.

Fonte: Próprio autor.

Além da tela inicial, conforme a Figura 3.1, todos os minigames possuem uma UI principal no canto superior esquerdo, como mostrado na Figura 3.2, na qual é mostrado o tempo que o jogador está no minigame, a quantidade de falhas e um botão de pausa.

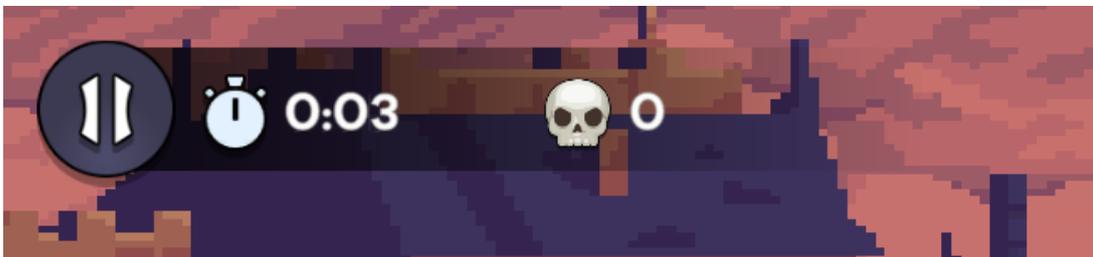


Figura 3.2: UI principal com tempo e quantidade de falhas.

Fonte: Próprio autor.

Ao pressionar o botão de pausa, uma nova UI é exibida na tela, conhecida como menu de pausa.

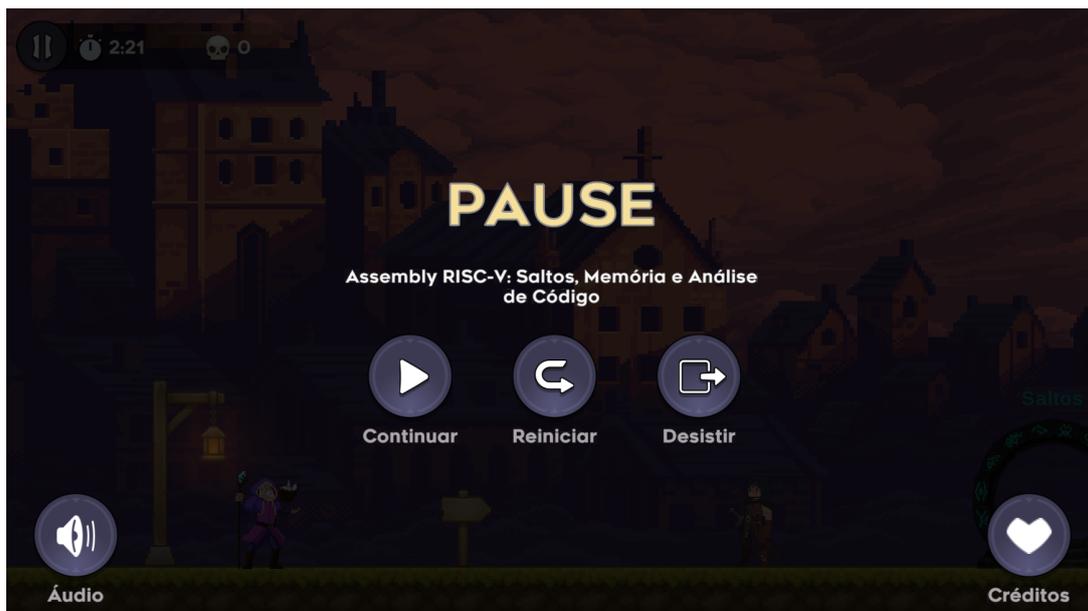


Figura 3.3: Menu de Pausa.

Fonte: Próprio autor.

Conforme ilustrado na Figura 3.3, a UI de pausa apresenta um texto descritivo do minigame a respeito dos conteúdos abordados. Além disso, conta com cinco botões distintos: um para retornar ao jogo, outro para reiniciar o minigame desde o início, um terceiro para desistir e voltar à tela inicial, um quarto para desativar a música principal do jogo e um quinto para exibir os créditos. Clicando no botão “Créditos”, a UI de créditos é aberta.

Figura 3.4: UI de Créditos.

Fonte: Próprio autor.

A UI de créditos pode ser visualizada na Figura 3.4. Nessa UI, são apresentados os créditos para todos os criadores de *assets* utilizados no projeto. É possível avançar pelos créditos clicando no botão “Próximo” ou esperar pelo avanço automático, que ocorre após um determinado tempo. Se um crédito se referir a um *asset* de imagem com animação, a imagem nos créditos também será animada.

Ao concluir um minigame, o jogador recebe o *feedback* de seu desempenho através da UI de conclusão.

Figura 3.5: UI de Conclusão dos minigames

Fonte: Próprio autor.

A UI, mostrada na Figura 3.5, apresenta inicialmente três estrelas desativadas. Ao completar o minigame, as estrelas são ativadas uma por uma, em sequência. No entanto, para que as três estrelas sejam ativadas, o jogador precisa ter um ótimo desempenho no jogo². Se o jogador tiver um desempenho médio, apenas duas estrelas serão ativadas. Em caso de um desempenho ruim, somente uma estrela será ativada. Se o desempenho for muito ruim, nenhuma estrela será ativada. Quando as três estrelas são ativadas, um efeito de explosão de estrelas é acionado.

As subseções a seguir descrevem cada minigame construído durante o desenvolvimento da pesquisa. Cada minigame é apresentado de forma linear, onde o texto é desenvolvido de forma a seguir o trajeto principal planejado. No processo, as mecânicas construídas e os desafios, elemento onde o jogador propostos são descritos.

²As estrelas refletem a pontuação do aluno no minigame. Cada estrela representa 33,3% da pontuação total.

3.2.1 Minigame 1.1

O primeiro minigame aborda os conceitos iniciais de Sistemas Computacionais e apresenta um pequeno tutorial para familiarizar o jogador, o qual pode ser visualizado na Figura 3.6.



(a) Primeiro painel do tutorial.



(b) Segundo painel do tutorial.



(c) Terceiro painel do tutorial.



(d) Quarto painel do tutorial.



(e) Quinto painel do tutorial.

Figura 3.6: Tutorial inicial.

Fonte: Próprio autor.

A Figura 3.6 informa ao jogador como se mover, sobre as informações na tela e sobre a importância de não trapacear no jogo.

Terminado as instruções iniciais, a movimentação do jogador é liberada, dando início a sua jornada. Na Figura 3.7, é possível observar a aparência visual do jogo, com seu personagem principal, o *Mago*. Logo a frente há uma personagem secundária, a *Donzela*, que é responsável por explicar os desafios ao jogador. Além disso, podemos notar a

presença de diversos objetos e uma iluminação projetada, que contribuem para a atmosfera envolvente do jogo.



Figura 3.7: Cenário inicial do minigame 1.1.

Fonte: Próprio autor.

Ao interagir com a Donzela, mostrado na Figura 3.8, a mesma fornece ao jogador informações sobre como conseguir superar o primeiro desafio, que envolve o conteúdo de *Álgebra Booleana*.



Figura 3.8: Diálogo entre jogador e Donzela.

Fonte: Próprio autor.

O desafio consiste em três estágios³, cada um com um altar de fogo que apresenta um problema ao jogador, e várias tochas que exibem uma solução que pode estar certa ou errada. O jogador deve escolher a tocha que torna a expressão do altar verdadeira e, assim, poder atravessar a “porta”, tal ação é feita pressionando a tecla “E” próximo à tocha. Caso o jogador escolha a tocha incorreta e tente atravessar a “porta”, seu personagem morre e o desafio recomeça.

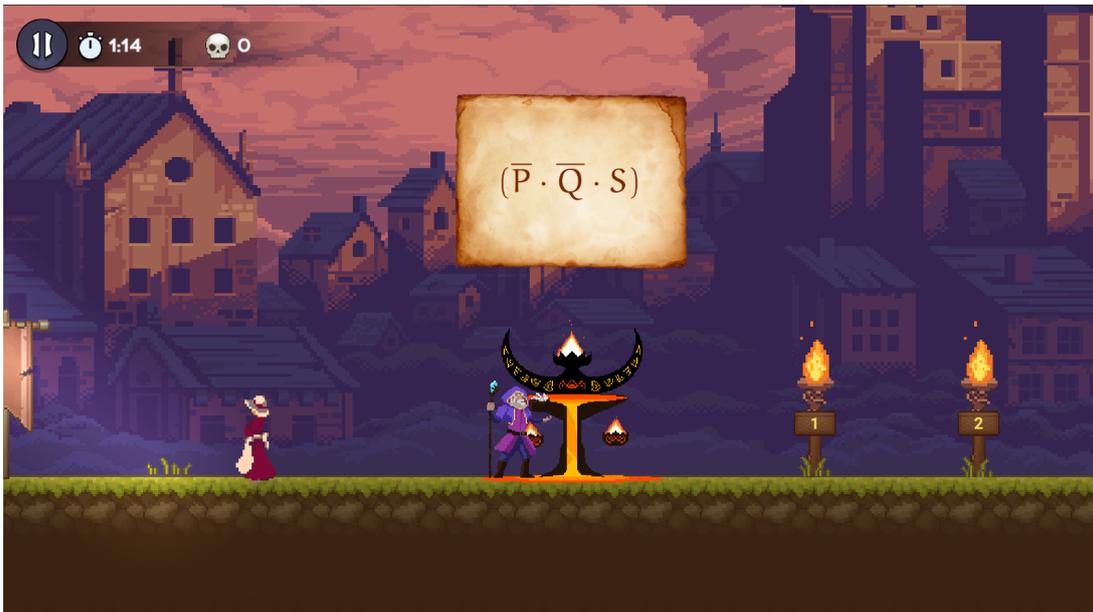


Figura 3.9: Primeiro estágio do desafio sobre álgebra booleana.

Fonte: Próprio autor.

Nos dois primeiros estágios do desafio, o altar de fogo exibe uma expressão booleana com variáveis, conforme ilustrado na Figura 3.9. O jogador precisa analisar atentamente a expressão booleana e determinar os valores booleanos (**0** ou **1**) das variáveis que tornam a expressão verdadeira.

Ao se aproximar de uma tocha, uma solução é exibida ao jogador, como ilustrado na Figura 3.10. Essa solução é formada pelas variáveis e seus respectivos valores booleanos que compõem o problema apresentado pelo altar de fogo. O jogador precisa analisar as opções disponíveis e escolher a tocha que possui as variáveis com os valores booleanos corretos, de modo a tornar a expressão booleana do altar de fogo verdadeira.

³A palavra “estágio” é utilizada no texto para indicar uma etapa de um determinado desafio. Ao mudar de estágio, o desafio tende a se tornar mais difícil.



Figura 3.10: Tocha do primeiro estágio do desafio sobre álgebra booleana.

Fonte: Próprio autor.

Ao escolher uma tocha, é apresentado um efeito visual de fogo em volta do jogador ao mesmo tempo que a chama da tocha se apaga, como mostrado na Figura 3.11. Dessa forma, o jogador “carrega” o fogo da tocha consigo.

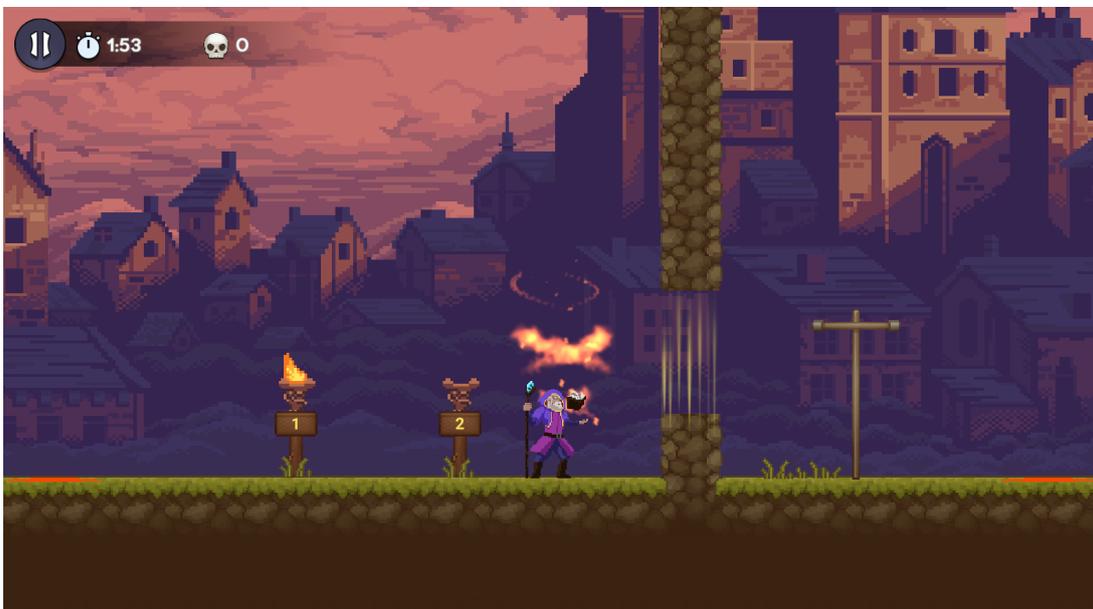


Figura 3.11: Jogador pega o fogo da tocha para avançar no desafio.

Fonte: Próprio autor.

No terceiro estágio do desafio, o altar de fogo exibe uma expressão booleana, mas com um símbolo de equivalência ao fim da expressão, como mostrado na Figura 3.12.

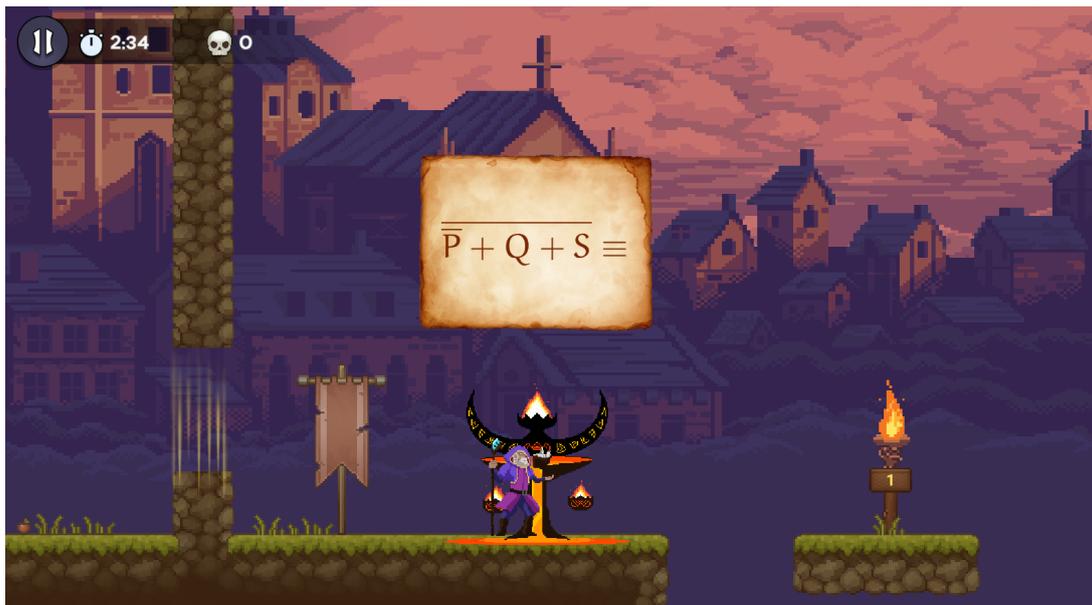


Figura 3.12: Terceiro estágio do desafio sobre álgebra booleana.

Fonte: Próprio autor.

A Figura 3.13 apresenta um novo estágio do desafio. Nesse estágio, o jogador precisa identificar a tocha com a expressão equivalente a do altar, para isso, o estudante tem de aplicar as equivalências de De Morgan.



Figura 3.13: Tocha do terceiro estágio do desafio sobre álgebra booleana.

Fonte: Próprio autor.

Após completar os três estágios do primeiro desafio, o jogador encontra um novo personagem, o *Cavaleiro Trapaceiro*, que apresenta o próximo desafio relacionado à *conversão*

de bases.

O Cavaleiro Trapaceiro possui o seguinte trecho de texto como diálogo: “Eu preciso de alguns pássaros especiais. Mas preciso apenas dos pássaros corretos. Eu soube que você sabe converter números de uma base qualquer para outra base qualquer. Isso será muito útil aqui. Eu tenho anotado aqui alguns números especiais e preciso que você os converta para a base que irei lhe falar. Existe um pássaro correspondente ao resultado encontrado, eu preciso que você o capture”. Na Figura 3.14 é mostrado um pouco desse diálogo. Em seguida, o Cavaleiro Trapaceiro diz que irá auxiliar o jogador e o instrui a ir até o círculo mágico.

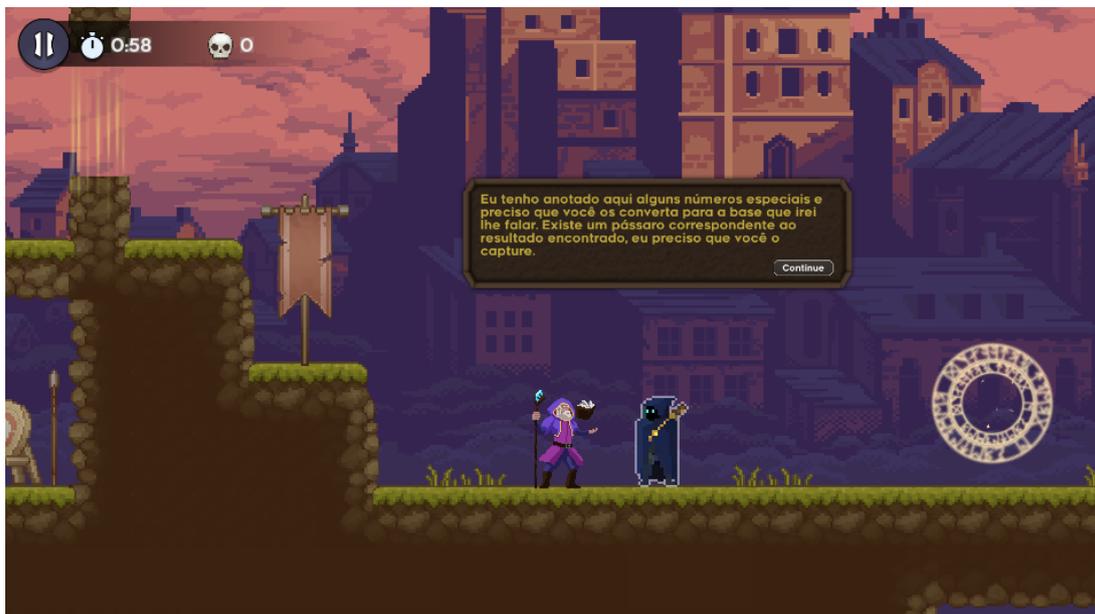


Figura 3.14: Diálogo entre jogador e Cavaleiro Trapaceiro.

Fonte: Próprio autor.

Ao entrar no círculo mágico, a câmera se desloca para o local onde o novo desafio acontece, como mostrado na Figura 3.15.

O Cavaleiro Trapaceiro informa ao jogador o número especial em decimal e a base numérica para a qual o jogador deve converter esse número. Com essas informações, o aluno deve clicar no pássaro que contenha o número especial convertido para a base especificada.



Figura 3.15: Cavaleiro Trapaceiro informando o número e a base ao jogador.

Fonte: Próprio autor.

Após a câmera estar posicionada, pássaros começam a aparecer, cada um com uma placa que exibe um número na base determinada, como ilustrado na Figura 3.16. Os pássaros são gerados em posições e direções aleatórias, podendo surgir tanto na direita quanto na esquerda da tela, e seguem um movimento retilíneo horizontal ou diagonal.

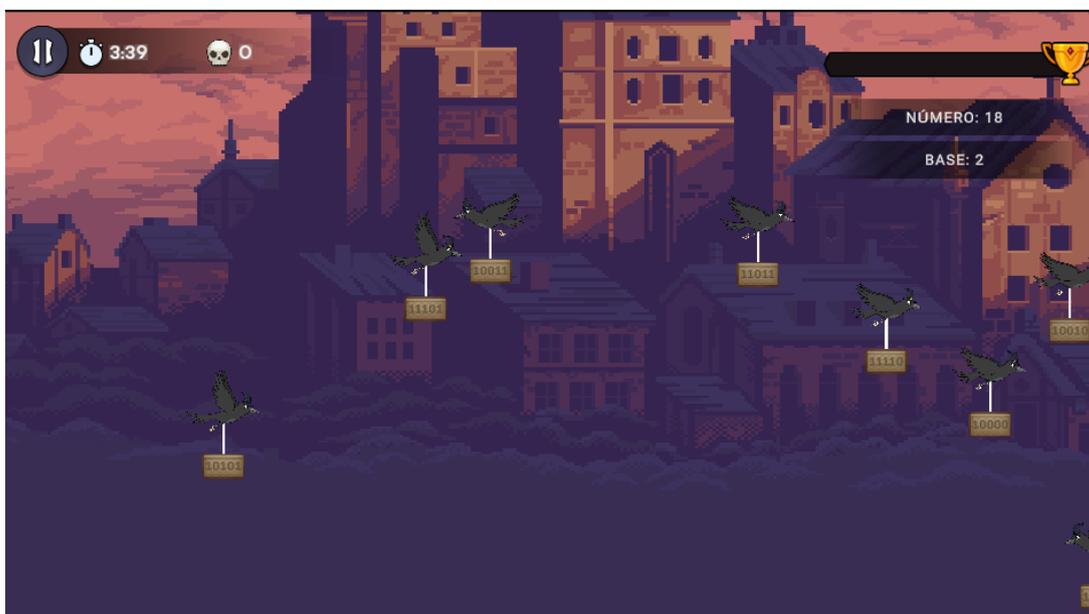


Figura 3.16: Pássaros gerados automaticamente com números na base determinada.

Fonte: Próprio autor.

O desafio necessita que o jogador capture ao todo cinco pássaros. Para que a dificuldade seja progressiva, os três primeiros números estão nas bases 2, 8 e 16 respectivamente. Enquanto os dois últimos números solicitados estão em bases aleatórias.

Concluindo o desafio, o *Cavaleiro Trapaceiro* realiza uma animação de movimento que o leva até um local próximo, como mostrado na Figura 3.17, e explica o próximo desafio ao jogador. O novo desafio envolve o conteúdo de operações simples entre bases distintas, ou seja, operações de soma, subtração e multiplicação entre números de qualquer base.

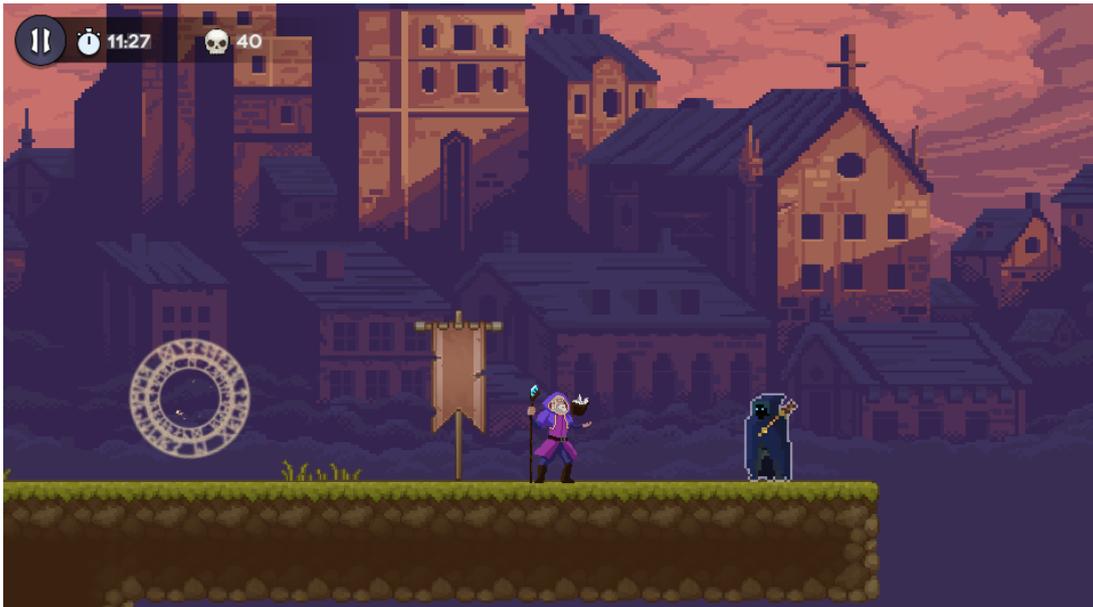


Figura 3.17: Posição final do Cavaleiro Trapaceiro.

Fonte: Próprio autor.

Este é o último desafio do primeiro minigame e se baseia em elevadores mágicos. O *Cavaleiro Trapaceiro* explica ao jogador o motivo de ter capturado os pássaros no desafio anterior, pois esses servirão para auxiliar o jogador nesse novo desafio.

Como ilustrado na Figura 3.18, o desafio é composto por quatro pássaros, localizados na região central da figura, e cinco elevadores, um onde jogador se encontra e mais 4 na região inferior da figura. Os pássaros à esquerda do jogador representam a operação que o jogador deve realizar. O primeiro e o terceiro pássaro exibem duas placas cada, a de cima especifica o número e a de baixo a base numérica. O segundo pássaro representa o tipo de operação, podendo ser soma, subtração ou multiplicação. O pássaro à direita do jogador está representado o resultado da operação no número X na base numérica informada.



Figura 3.18: Último desafio do primeiro minigame.

Fonte: Próprio autor.

Em relação aos elevadores, os quatro elevadores alinhados na parte inferior da Figura 3.18 exibem placas com números na base especifica pelo pássaro com o número X . Apenas um desses números é o resultado correto da operação indicada. O jogador precisa saltar no elevador com o valor correto para a operação. Caso escolha o elevador incorreto, paredes com lanças surgem, o jogador morre e o desafio é reiniciado.

Ao pular no elevador correto, o mesmo sobe e centraliza na tela, enquanto os outros elevadores saem de cena e, em seguida, retornam com novos valores e uma nova operação. Visualmente, a situação se assemelha à mostrada na Figura 3.18. Esse processo se repete por cinco vezes, totalizando cinco operações.

Após concluir todas as operações, o jogador é transportado pelo elevador mágico até o local do próximo desafio. Nesse momento, o minigame é encerrado e a tela de conclusão é apresentada, semelhante a Figura 3.5. Nessa tela são apresentadas estrelas de 1 à 3, refletindo o desempenho do aluno no minigame, e um resumo dos conceitos exercitados.

3.2.2 Minigame 1.2

O aluno conclui o minigame 1.1 tendo os primeiros contatos com dinâmicas envolvendo Álgebra Booleana, e conversão de base e operações aritméticas envolvendo números em bases distintas. O minigame 1.2 dá continuidade a essas dinâmicas, apresentando novas representações visuais, interações e incluindo novos conteúdos, como ponto fixo, acesso a memória, complemento de base e ponto flutuante.

O estudante chega com mais experiência no minigame 1.2 do que em relação ao minigame anterior. Dessa forma, os desafios do minigame 1.2 são mais difíceis para manter o exercício instigante [31] e incitar o sentimento de progressão [1].

O primeiro desafio, o *Desafio dos Potes*, trabalha com o conceito de ponto fixo. Essa é uma notação onde um número racional r é representado por valor inteiro x , e seu *scaling factor* s , o número pelo qual x tem de ser multiplicado para restaurar seu valor original r [32].

O Desafio dos Potes consiste no jogador encher os potes de forma a representar o valor requisitado. Para iniciar o desafio, o estudante interage com a Donzela, como mostrado na Figura 3.19, o qual vai instruí-lo como funciona a dinâmica do desafio e informar o valor desejado a ser inserido nos potes. A representação visual consiste nos níveis de água nos potes, cada pote representa um dígito, os números são representados da maior para a menor potência (lendo da esquerda para a direita).



Figura 3.19: Donzela apresentando o Desafio dos Potes.

Fonte: Próprio autor.

A Figura 3.20 mostra o momento que o desafio começa, com todos os potes vazios. O jogador precisa preenchê-los com água conforme instruído anteriormente, conforme Figura 3.19.

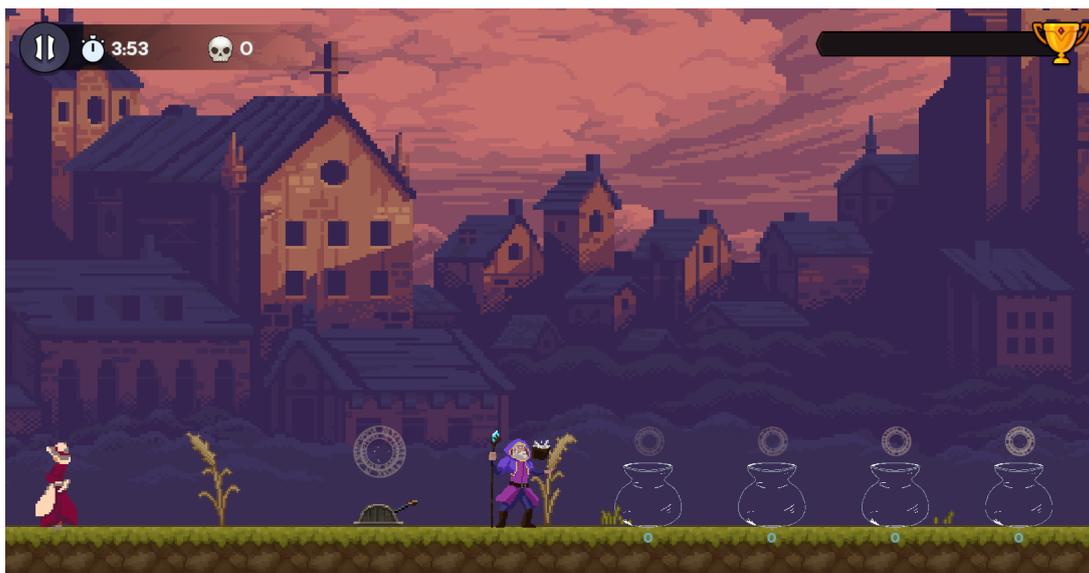


Figura 3.20: Início do Desafio dos Potes.

Fonte: Próprio autor.

O jogador deve interagir com cada pote de modo a inserir o dígito do número 203,47 na base 6 considerando 2 casas fracionárias, conforme instruído pela Donzela na Figura 3.19. A Figura 3.21 mostra os potes preenchidos com os valores corretos.

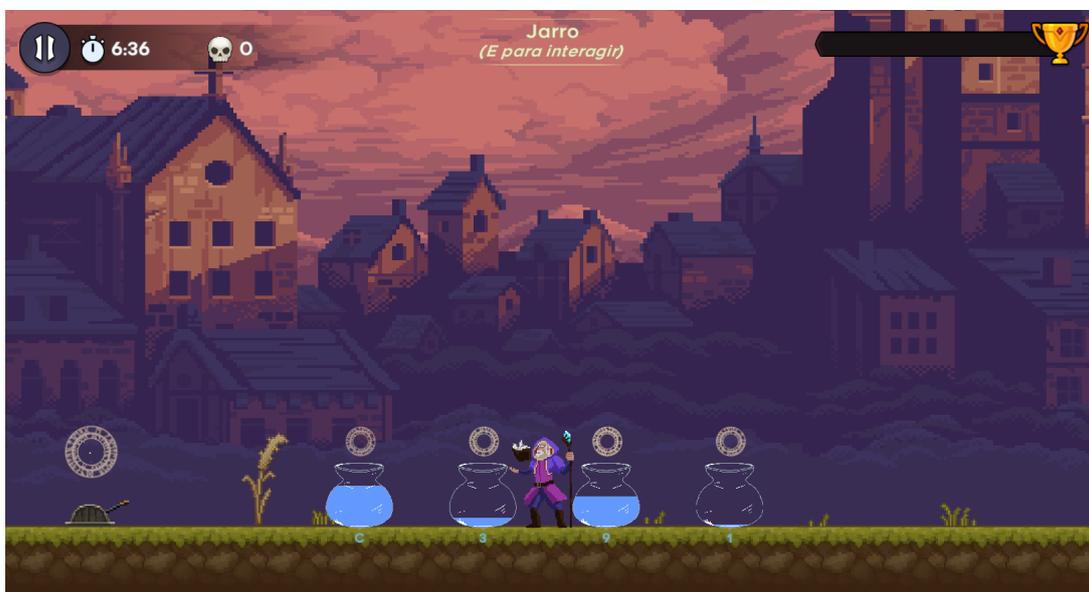


Figura 3.21: Jogador preenchendo os potes para o valor especificado pela *Donzela*.

Fonte: Próprio autor.

Quando o jogador estiver certo de sua resposta ele deve acionar a alavanca, localizada a esquerda do primeiro pote.

Após o jogador confirmar sua resposta, o jogo realiza a validação. Se caso a resposta estiver correta, um emblema surge no centro da tela sinalizando e a barra de progresso é incrementada (visível na região superior direita da tela), conforme a Figura 3.22. Caso o jogador erre, uma falha é contabilizada, a barra de progresso se esvazia e o desafio recomeça.

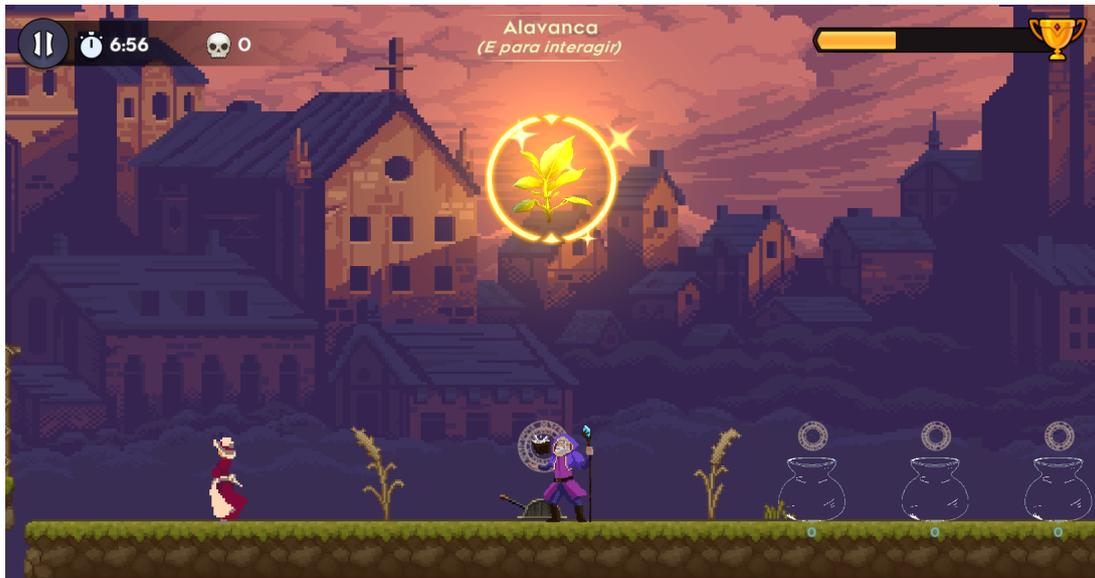


Figura 3.22: Efeito audiovisual ao jogador acertar a resposta do Desafio dos Potes.

Fonte: Próprio autor.

Ao acertar a resposta, o desafio vai para o próximo estágio. O Desafio dos Potes é constituído por três estágios, com progressão de dificuldade. A cada estágio, há um incremento de casas fracionárias a serem consideradas na resposta. No primeiro estágio são duas casas fracionárias e no terceiro e último estágio são seis.

Completando as três etapas do Desafio dos Potes, um elevador surge no canto direito da tela com destino ao local do próximo desafio, como ilustrado na Figura 3.23. Para utilizá-lo basta que o jogador interaja com ele.



Figura 3.23: Elevador à direita do jogador para levá-lo ao próximo puzzle.

Fonte: Próprio autor.

Após pegar o elevador e seguir adiante por um tempo o jogador encontra um novo personagem, o *Senhor*, conforme a Figura 3.24. Ao interagir com o Senhor, o jogador é introduzido a um novo desafio, o *Desafio das Prateleiras*.

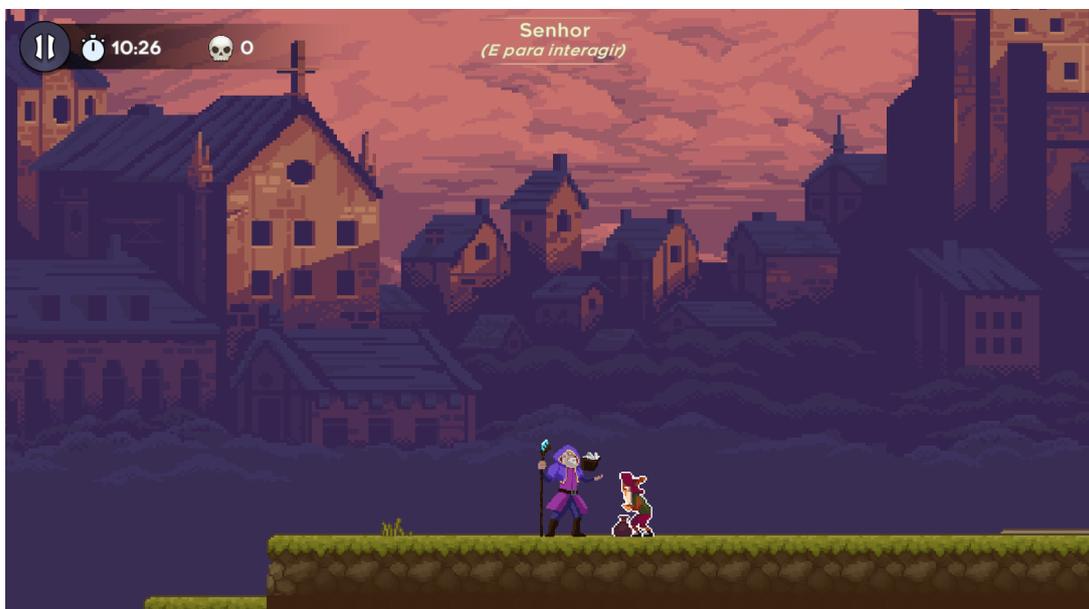


Figura 3.24: Cenário antes do Desafio das Prateleiras.

Fonte: Próprio autor.

O Desafio das Prateleiras é voltado para a prática de conversão de bases, onde o jogador deve informar os valores armazenados em uma determinada posição da prateleira. Logo,

o desafio também aplica conceitos de acesso a memória e ponteiros. Cada prateleira é rotulada com um valor de 32 bits em hexadecimal, tal valor representa um endereço de memória. Na prateleira, existem 8 baús, cada baú representa 1 *word*, ou 32 bits. Ou seja, se uma prateleira tem endereço 0x00000A40, o endereço do primeiro baú começa nessa posição, o do segundo baú começa no endereço 0x00000A44 e assim por diante, até chegar no oitavo baú com endereço 0x00000A56, e, a próxima prateleira tem endereço 0x00000A60.

Na Figura 3.25, o *Senhor* instrui o estudante a buscar o baú localizado no endereço de memória 0x00000A50 e informar o valor contido nesse baú na base 10. Os valores armazenados nos baús podem estar armazenados em bases que podem ser desde a base 5 até a base 36, sendo que esse valor é gerado aleatoriamente.

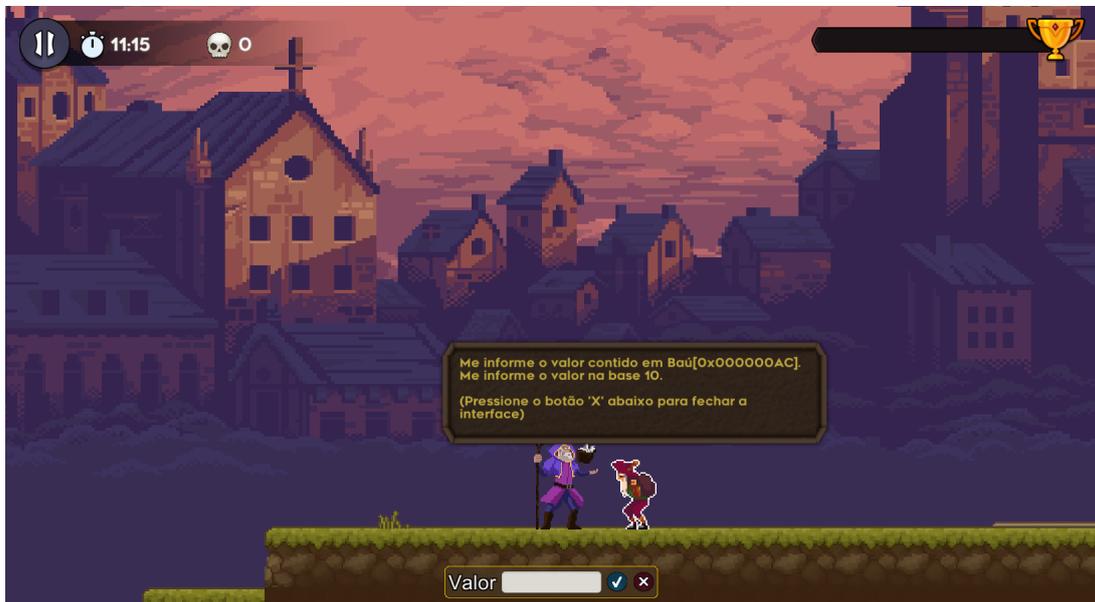


Figura 3.25: *Senhor* explicando o que deve ser feito.

Fonte: Próprio autor.

O jogador deve interagir com o baú na posição instruída, converter o valor 51 para a base 10, conforme a Figura 3.26a, e responder no campo de entrada do diálogo com o *Senhor*, conforme a Figura 3.26b. Ao responder corretamente, o jogador avança para o próximo estágio do desafio, o qual é um pouco mais difícil que o anterior. A mudança de dificuldade nesse desafio se dá pela alteração de redirecionamentos que o jogador tem de seguir até encontrar o valor desejado.



(a) Baú contém o valor 51 (na base 8).



(b) Jogador inserindo resposta (valor 41).

Figura 3.26: Jogador buscando a resposta e retornando ao *Senhor*.

Fonte: Próprio autor.

O desafio começa sem redirecionamento algum e, a cada estágio é incrementado em 1 a quantidade de redirecionamento. Como o desafio tem três estágios, no terceiro o jogador tem dois redirecionamentos antes de chegar no valor desejado, no qual se aplica a ideia de ponteiros, onde o endereço do valor desejado é armazenado em outro endereço e assim sucessivamente.

A Figura 3.27 mostra o personagem *Senhor* apresentando o terceiro estágio do desafio, onde é possível notar que o valor de um baú é utilizado para acessar o valor de outro. Em outras palavras, o aluno precisa “seguir” o endereço informado nos dois primeiros baús até chegar no endereço do baú final, que contém o valor a ser retornado para o *Senhor*.



Figura 3.27: *Senhor* apresentando terceiro estágio do desafio.

Fonte: Próprio autor.

A ideia por trás da aplicação dos conceitos de acesso a memória e ponteiros é de familiarizar o estudante e estimulá-lo às suas primeiras práticas com esses assuntos. Dessa forma, o mesmo irá adquirir uma certa facilidade em compreender e depurar o comportamento de códigos *Assembly* ou na linguagem de programação C.

Ao passar do terceiro estágio do Desafio das Prateleiras o caminho adiante será desbloqueado e o jogador conseguirá seguir adiante, onde se deparará com uma nova personagem secundária, a *Esgrimista*, como mostrado na Figura 3.28. Além de fornecer orientações sobre o último desafio desse minigame, a Esgrimista se tornará a adversária do jogador, convidando-o para um duelo um contra um.

desculpe o incômodo

Figura 3.28: Diálogo entre jogador e Esgrimista.

Fonte: Próprio autor.

Ao iniciar o duelo, a câmera centraliza nos dois atores⁴ e uma música energética começa a tocar. Nessa emocionante disputa, o jogador tem a oportunidade de aplicar seus conhecimentos em complemento de base e ponto flutuante para derrotar a Esgrimista e concluir o minigame.

Esse desafio é apresentado por meio de uma UI que exhibe questões relacionadas ao complemento de base, conforme a Figura 3.29. O jogador deve converter corretamente o número apresentado, que está no formato de sinal e magnitude. Ao realizar a conversão corretamente, a Esgrimista sofrerá dano perdendo um pouco de sua vida.

⁴Termo comumente utilizado para referenciar objetos dinâmicos em jogos.



Figura 3.29: Interface gráfica com a questão de Complemento de Base.

Fonte: Próprio autor.

Em seguida, a interface gerará uma nova questão, porém, sobre ponto flutuante, conforme a Figura 3.30, exigindo que o jogador faça a conversão correta para causar mais dano à Esgrimista. Esse ciclo se repete, alternando a geração de novas questões entre complemento de base e ponto flutuante.



Figura 3.30: Interface gráfica com a questão de Ponto Flutuante.

Fonte: Próprio autor.

A interface gráfica é composta por duas barras de vida, uma para o jogador e outra para a Esgrimista. Além disso, há uma barra na cor roxa, centralizada no canto inferior da tela, que representa o tempo para responder as questões propostas. Caso o jogador erre a questão ou o tempo se esgote, a Esgrimista realizará um ataque que causará dano ao jogador, diminuindo sua barra de vida. O desafio continua até que uma das barras de vida seja zerada. Se a barra de vida do jogador chegar a zero, ele perderá e terá que refazer esse último desafio.

3.2.3 Minigame 2.1

Este minigame marca o início do conteúdo de *Assembly* e foi projetado como uma introdução amigável ao módulo 2. Sua proposta é oferecer uma experiência mais fácil aos jogadores, com foco principal em operações aritméticas em *Assembly*.

O jogo começa em um ambiente iluminado com a presença da Donzela à frente do jogador, conforme a Figura 3.31. Ela explica que, nos desafios desse minigame, será necessário observar os valores do registrador $s0$.

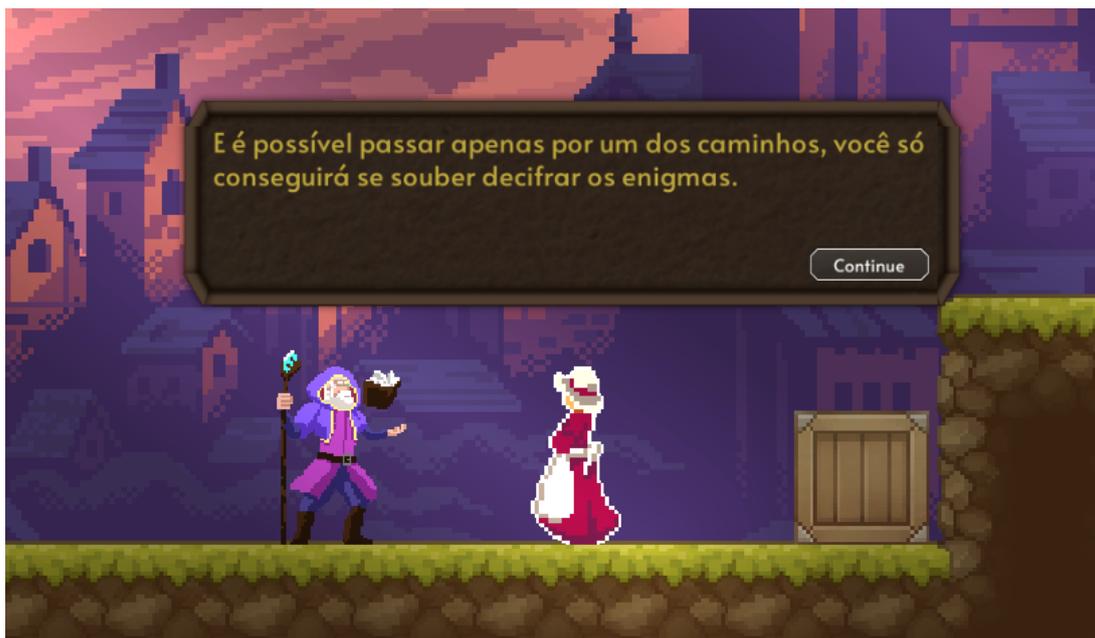


Figura 3.31: Diálogo entre jogador e Donzela.

Fonte: Próprio autor.

Em seguida, o jogador se depara com uma placa contendo um código em *Assembly* e três caminhos diferentes, conforme a Figura 3.32, sendo apenas um deles o correto. Cada caminho possui uma placa com um número, e o jogador deve analisar o código em *Assembly* e escolher o caminho cujo número corresponde ao valor do registrador $s0$.

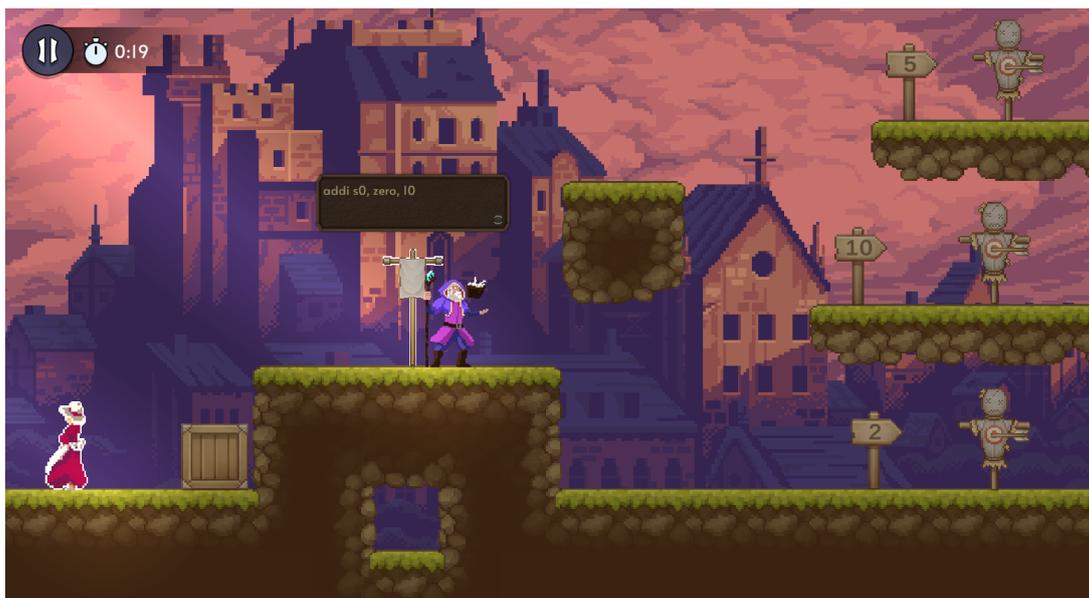


Figura 3.32: Primeiro e mais fácil desafio que o estudante enfrenta no minigame.

Fonte: Próprio autor.

Cada um dos caminhos possui uma armadilha, contendo uma lança. Caso o jogador escolha o caminho incorreto, a armadilha será acionada e a lança é disparada em direção ao jogador, fazendo com que o mesmo tenha de recomeçar o desafio.

Após escolher o caminho correto, o jogador avançará para o próximo desafio, que apresenta uma placa com um código em *Assembly*. Novamente, o jogador precisa fazer a escolha correta, mas desta vez deve pular na pedra correta. As pedras incorretas não possuem colisão, de modo que o jogador cairá no abismo e morrerá.

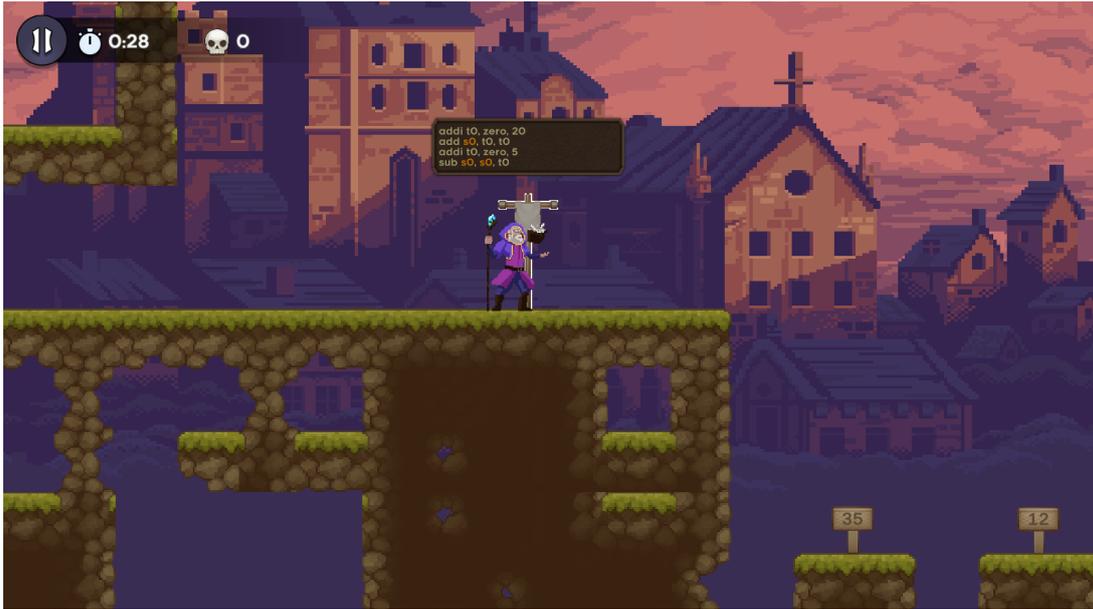
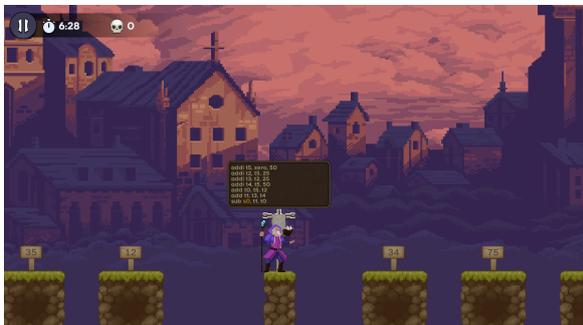


Figura 3.33: Segundo desafio do minigame.

Fonte: Próprio autor.

Esse desafio é composto por três estágios, nos quais a dificuldade aumenta progressivamente. Se o jogador falhar, ele retornará para a última placa de código *Assembly* lida. O primeiro estágio desse desafio pode ser visualizado na Figura 3.33.



(a) Segundo estágio do segundo desafio.



(b) Terceiro estágio do segundo desafio.

Figura 3.34: Jogador completando o segundo desafio.

Fonte: Próprio autor.

Neste desafio, o jogador tem a oportunidade de praticar as operações aritméticas em *Assembly*. O primeiro e segundo estágio, conforme apresentado nas Figuras 3.33 a 3.34a, são focados em soma e subtração, enquanto o terceiro estágio inclui as operações de multiplicação e divisão, conforme mostra a Figura 3.34b.

Como ilustrado na Figura 3.35, ao avançar, o jogador é surpreendido por um novo personagem secundário, o *Fantasma*. Esse é responsável por explicar o último desafio do minigame.

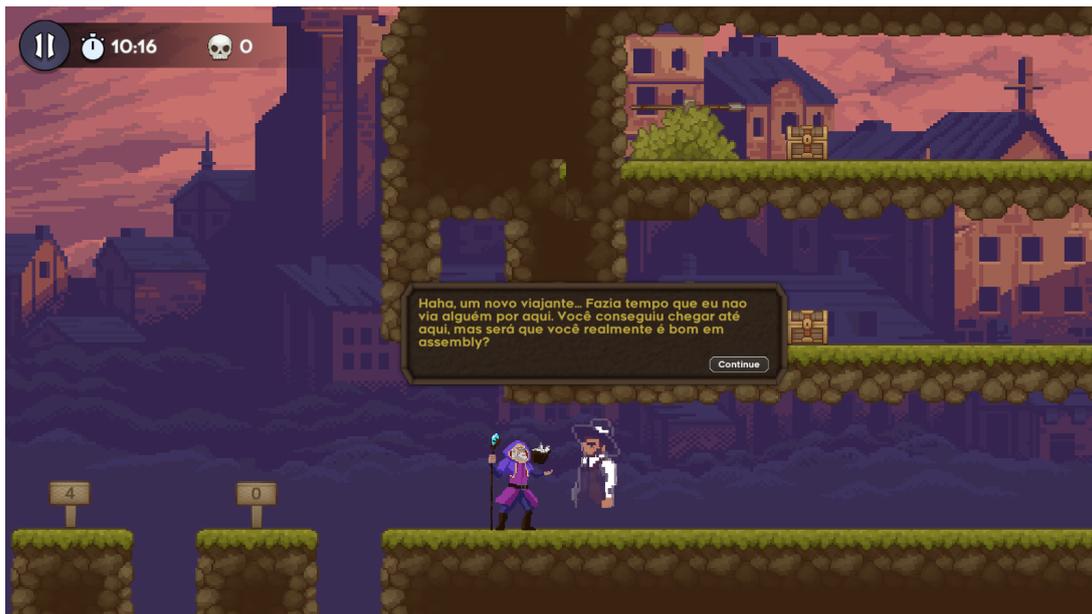


Figura 3.35: Diálogo com o personagem Fantasma.

Fonte: Próprio autor.

O Fantasma explica ao jogador que ele deve analisar cuidadosamente o código *Assembly* indicado pela placa à frente, e que a cada vez que o registrador *s0* recebe um valor, o jogador deve escolher o caminho correspondente e abrir o baú.

No entanto, o Fantasma alerta que a rapidez é essencial nesse desafio. Se o jogador demorar muito para resolver a tarefa, fantasmas inimigos começarão a aparecer e atrapalhar o seu caminho. Além disso, se o jogador abrir o baú errado, uma armadilha é acionada, disparando uma lança em sua direção e obrigando-o a recomençar o desafio.

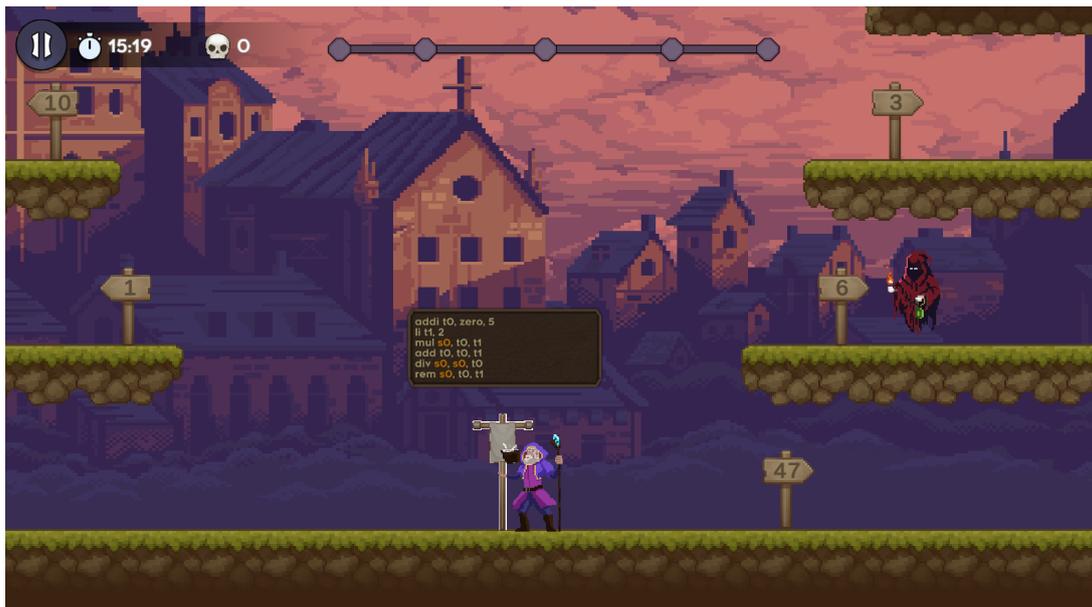


Figura 3.36: Jogador no último desafio e Inimigo patrulhando o caminho.

Fonte: Próprio autor.

Na Figura 3.36, é possível ver o fantasma que apareceu após o jogador demorar muito tempo para completar o desafio. Também é mostrada uma das cinco questões selecionadas do banco de questões para este desafio.

Os fantasmas inimigos patrulham o caminho onde os baús estão localizados. Quando o jogador se aproxima pela frente do fantasma, ocorre um movimento de ataque flutuante em direção ao jogador. Caso o ataque atinja o jogador, será necessário reiniciar o desafio.

Ao analisar os códigos *Assembly* e resolver o desafio por 5 vezes consecutivas, o jogador vence e finaliza o minigame, recebendo seu *feedback* final e sua nota no Moodle, semelhante a Figura 3.4.

3.2.4 Minigame 2.2

O minigame 2.2 se concentra no temas de deslocamento e operações booleanas em *Assembly*. O jogo começa com o jogador próximo a uma loja de frutas e de algumas pedras flutuantes que se movem para cima e para baixo de forma repetitiva, como mostrado na Figura 3.37.

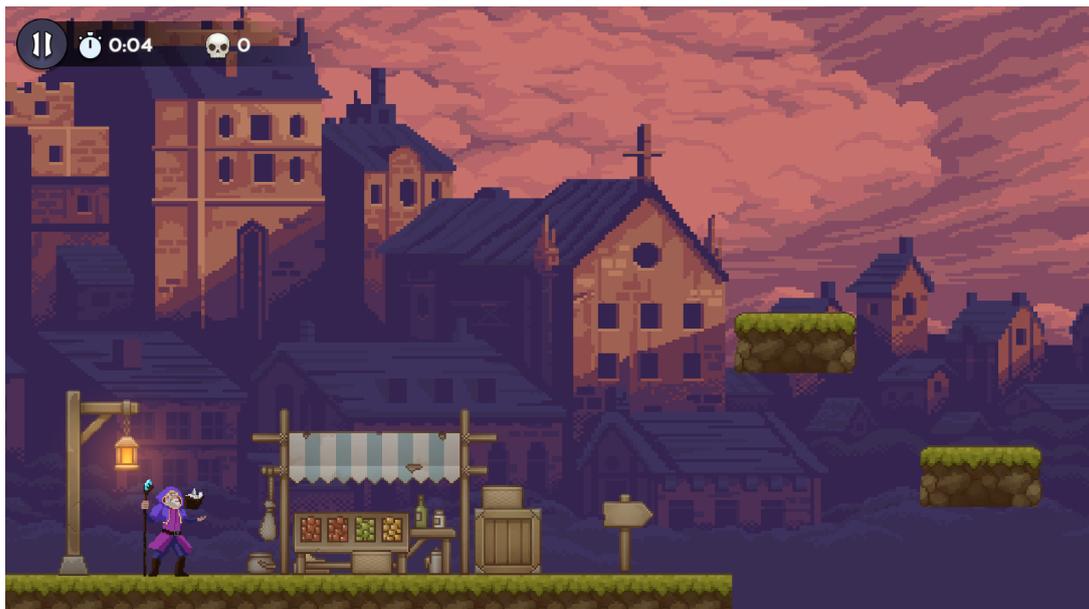


Figura 3.37: Início do minigame 2.2.

Fonte: Próprio autor.

Todo o cenário desse minigame foi projetado para transmitir a ideia de deslocamento, incentivando o jogador a associar esses movimentos ao conteúdo abordado sobre deslocamento em *Assembly*. Depois que o jogador atravessa as pedras flutuantes, ele se depara com o personagem Fantasma, conforme a Figura 3.38, responsável por explicar o primeiro desafio presente nesse minigame, o *Desafio da Ponte*.



Figura 3.38: Diálogo com o Fantasma no Desafio da Ponte.

Fonte: Próprio autor.

O Fantasma explica que jogador deve usar o conhecimento sobre deslocamento em *Assembly RISC-V* para criar uma passagem segura, permitindo que o jogador avance para o próximo desafio.

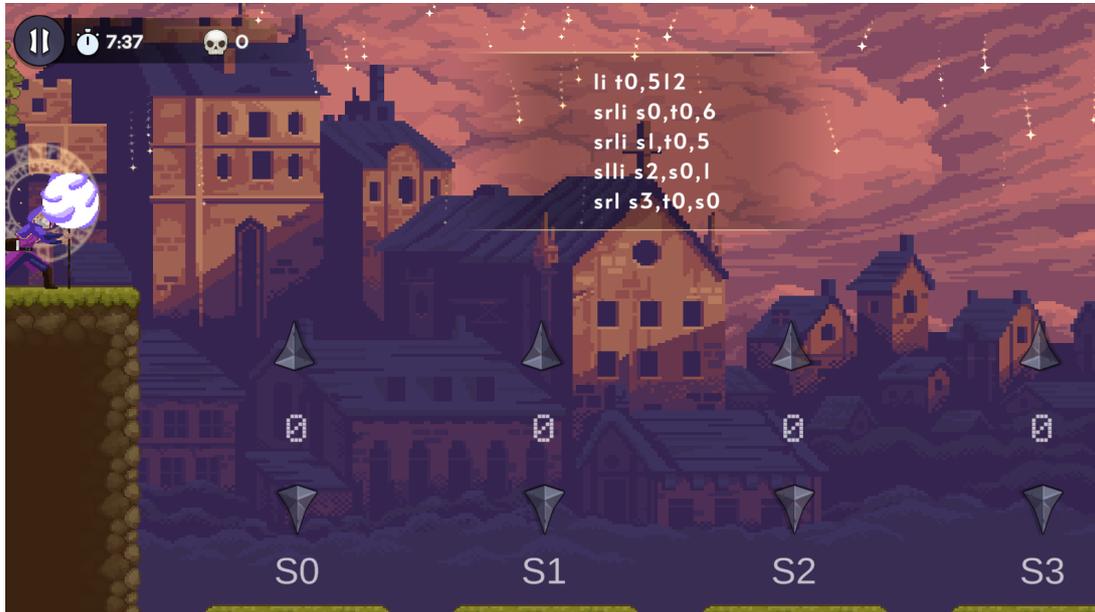


Figura 3.39: Jogador no Desafio da Ponte.

Fonte: Próprio autor.

No desafio em questão, ilustrado na Figura 3.39, o jogador se depara com um código *Assembly* contendo várias instruções de deslocamento. Sua tarefa é analisar o código e fornecer os valores corretos para os registradores $s0$, $s1$, $s2$ e $s3$. Assim que todos os valores forem corretamente inseridos, as pedras se deslocam para cima formando uma passagem, permitindo que o jogador avance com segurança para o outro lado.

Ao finalizar o desafio e atravessar a ponte formada, o jogador encontra novamente o Fantasma, conforme a Figura 3.40, que agora lhe auxilia sobre o *Desafio da Porta*.



Figura 3.40: Diálogo com o Fantasma no Desafio da Porta.

Fonte: Próprio autor.

Neste desafio o jogador precisa abrir uma porta que está com as trancas quebradas. Assim, a UI, ilustrada na Figura 3.41, é formada por duas chaves e quatro pinos. O jogador precisa deslocar os pinos da tranca para encaixar corretamente nas duas chaves e assim abrir a porta.



Figura 3.41: Desafio da Porta.

Fonte: Próprio autor.

Os pinos representam os registradores s_0 , s_1 , s_2 e s_3 , respectivamente. O jogador precisa informar os valores dos registradores t_0 , t_1 , t_2 e t_3 para que os pinos alcancem os encaixes das chaves. As instruções de deslocamento são mostradas abaixo dos pinos.

Após concluir o Desafio da Porta, o jogador é levado ao próximo desafio, conhecido como *Desafio dos Lasers*. Diferentemente dos desafios anteriores, neste não há um personagem específico para explicar o objetivo, pois as instruções são visualmente autoexplicativas.



Figura 3.42: Jogador no Desafio dos Lasers.

Fonte: Próprio autor.

No Desafio dos Lasers, mostrado na Figura 3.42, o jogador enfrenta a tarefa de deslocar a pedra flutuante para bloquear o feixe de laser e abrir caminho para sua passagem. Utilizando as instruções de deslocamento disponíveis, o jogador precisa posicionar estrategicamente a pedra de forma a obstruir o laser e, assim, avançar para a próxima posição segura. Esse processo é repetido até que o jogador tenha superado completamente o Desafio dos Lasers.

Prosseguindo, o jogador se depara com o encontro de um novo personagem, o *Viajante*. Este personagem explica ao jogador sobre o próximo desafio, o *Desafio dos Fantasmas*.

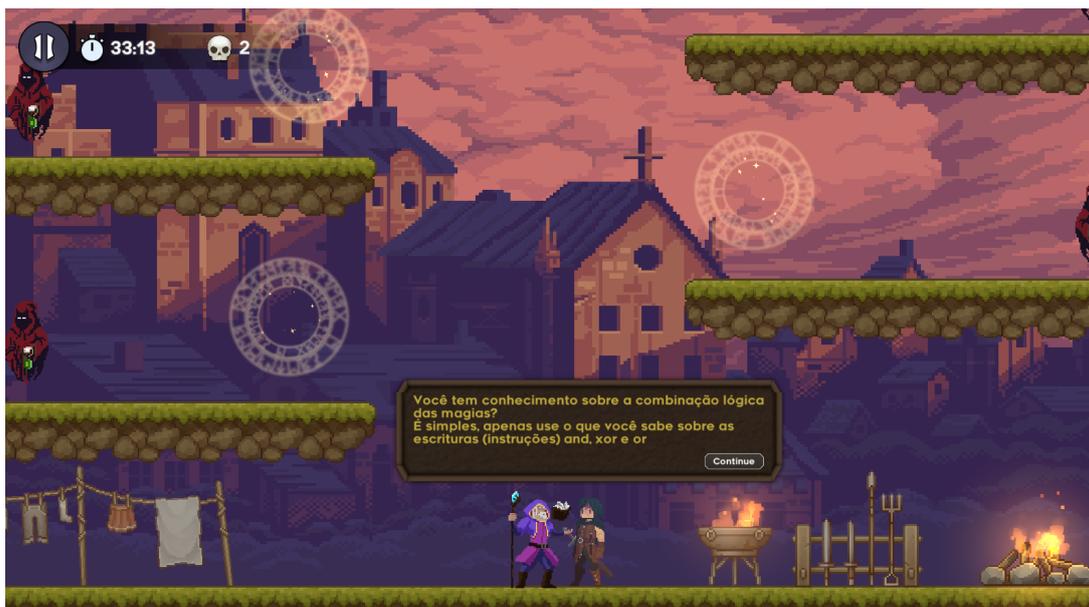


Figura 3.43: Desafio dos Fantasmas.

Fonte: Próprio autor.

Na Figura 3.43, o Viajante explica que os fantasmas criaram uma barreira invisível que impede qualquer um de avançar, e que para quebrar essa barreira é necessário derrotar os fantasmas. Assim, o jogador precisa utilizar o conhecimento das instruções de operações lógicas, como *and*, *or* e *xor*, para realizar uma magia efetiva contra cada um dos fantasmas.



Figura 3.44: Jogador utilizando o círculo mágico para derrotar o fantasma.

Fonte: Próprio autor.

Como mostrado na Figura 3.44, ao adentrar o círculo mágico, o jogador se depara com um código em *Assembly* contendo instruções de operações lógicas. O objetivo é determinar corretamente os valores armazenados nos registradores *s0* e *s1* após a execução do código. Se o jogador fornecer o valor correto, o fantasma é derrotado, porém, se a resposta estiver

incorreta, o fantasma desperta e começa a patrulhar. Caso o jogador seja atacado com sucesso pelo fantasma, será necessário reiniciar o desafio.

Após derrotar todos os fantasmas e desbloquear o caminho, o jogador reencontra o Viajante. Esse então apresenta o desafio final do minigame, o *Desafio do Necromante*. Conforme mostra a Figura 3.45.



Figura 3.45: Diálogo com o Viajante no Desafio do Necromante.

Fonte: Próprio autor.

No Desafio do Necromante, o jogador precisa aplicar seus conhecimentos sobre instruções de deslocamento e operações lógicas para derrotar um poderoso inimigo, o *Necromante*.

Como mostrado na Figura 3.46, o Necromante avança em direção ao jogador, destruindo obstáculos em seu caminho e lançando esferas mágicas em direção aos céus. O *Domo Protetor* mantém o jogador protegido desses ataques aéreos. Caso o jogador não esteja protegido, as esferas mágica do Necromante podem atingi-lo. Caso o Necromante alcance o jogador ou o acerte com uma esfera mágica, o mesmo morre e será necessário tentar superar o desafio novamente.

Figura 3.46: Necromante avançando sobre os obstáculos.

Fonte: Próprio autor.

Para realizar um ataque no Necromante, o jogador deve analisar o código *Assembly*, que possui instruções de operações lógicas, no canto inferior direito da tela e informar os valores corretos dos registradores *s0* e *s1*. Ao confirmar, se o valor estiver correto, o Necromante sofrerá dano e perderá um pouco de sua vida.

Durante o ataque, conforme a Figura 3.46, a câmera principal foca no Necromante e uma câmera secundária mantém o jogador atualizado sobre o que está acontecendo próximo ao seu personagem, isso é feito através de um painel miniatura localizado no canto superior direito da tela. Quando o jogador está em modo de defesa, ao clicar no painel de resposta na região inferior esquerda, a câmera principal alterna com a secundária e o foco principal fica no jogador e o painel miniatura mostra o Necromante, conforme a Figura 3.47.



Figura 3.47: Câmera focando o jogador no Domo Protetor.

Fonte: Próprio autor.

Caso o Domo Protetor seja atingido por uma bola de magia, o mesmo se quebra, deixando o jogador sem proteção. Para deslocar e arrumar o domo, o jogador precisa analisar o código *Assembly*, que possui instruções de deslocamento, no canto inferior esquerdo da tela, representado pela palavra Defesa, e informar os valores corretos para os registradores $s0$ e $s1$.

O jogador precisa coordenar suas respostas de tal forma que o Domo Protetor o proteja de ataques do Necromante e ao mesmo tempo precisa realizar ataques ao Necromante para impedir que ele se aproxime. O jogador vence esse desafio quando a vida do Necromante for igual a zero.

3.2.5 Minigame 2.3

O minigame 2.3 inicia com o Viajante diante de um portal, posicionado logo à frente do jogador, conforme a Figura 3.48. O Viajante introduz o desafio principal do minigame, o *Desafio dos Portais* localizado no *Vale dos Portais*. Esse desafio envolve a aplicação dos conceitos de saltos condicionais e incondicionais. Após interagir com o Viajante e receber as instruções, o jogador é instruído a entrar no portal de saltos.

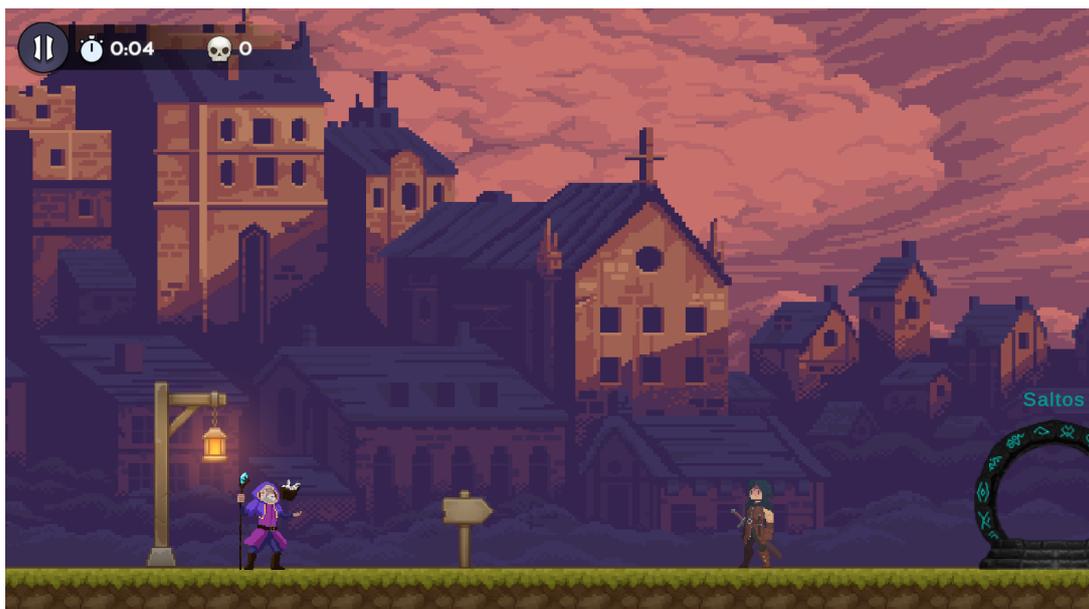


Figura 3.48: Cenário inicial do minigame 2.3.

Fonte: Próprio autor.

O jogador é transportado para o Vale dos Portais toda vez que atravessa um portal de salto, conforme a Figura 3.49. Ao chegar nesse local, o jogador se depara com cinco portais distintos, cada um deles identificado por um nome específico.

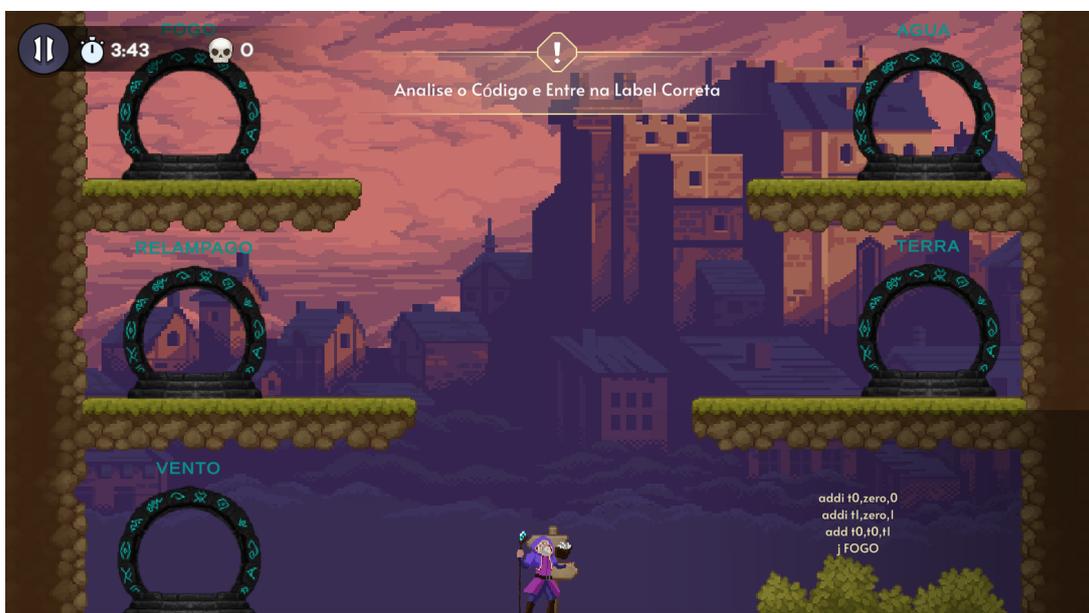


Figura 3.49: Vale dos Portais no Desafio dos Portais.

Fonte: Próprio autor.

No canto inferior direito da tela é exibido o código *Assembly* das instruções de saltos condicionais e incondicionais. O desafio consiste em analisar o código e determinar para

qual rótulo (do Inglês, *label*) o salto será executado, seja ele condicional ou incondicional. Com base nessa análise, o jogador deve selecionar o portal correspondente à *label* identificada.

Caso o jogador selecione o portal errado, fantasmas inimigos surgem no cenário, e o jogador tem de recomeçar o desafio. No entanto, ao entrar no portal correto, o jogador realiza um salto no espaço-tempo do jogo e é transportado para uma nova localização, onde se depara com um desafio completamente novo. Nesse minigame, o jogador terá de completar o Desafio dos Portais diversas vezes.

Após atravessar o Vale dos Portais pela primeira vez, o jogador é transportado para uma região onde algumas rochas estão caídas, cada uma com um baú em cima identificado por um número hexadecimal, conforme a Figura 3.50a. Mais à frente, o Viajante aguarda o jogador. As rochas caídas se erguem a medida que o jogador se aproxima delas, conforme a Figura 3.50b.



(a) Rochas caídas.



(b) Rochas elevadas.

Figura 3.50: Dinâmica de elevação de rochas com proximidade do jogador.

Fonte: Próprio autor.

Como mostrado na Figura 3.51, o Viajante explica ao jogador sobre o que se trata os baús e solicita sua ajuda, dando início ao *Desafio do Acesso à Memória*.

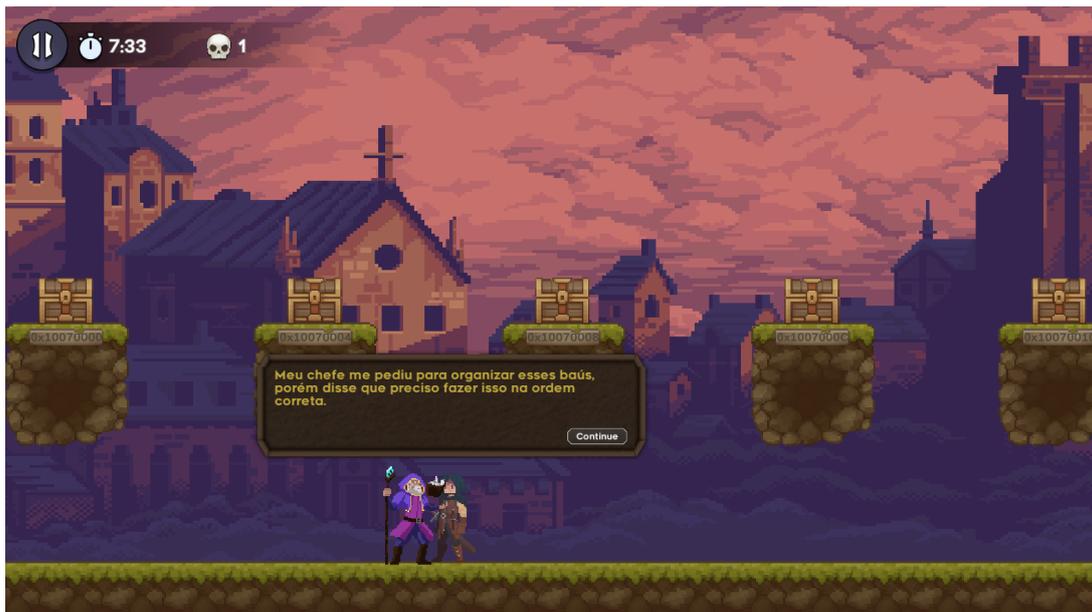


Figura 3.51: Desafio do Acesso à Memória

Fonte: Próprio autor.

No Desafio do Acesso à Memória, o jogador precisa ajudar o Viajante na organização dos conteúdos armazenados nos baús. A tarefa consiste em transferir corretamente o conteúdo de um baú para outro, ou seja, remover o que está dentro de um baú e guardar no baú adequado.

Para cumprir esse objetivo, o jogador deve analisar o código *Assembly*, o qual possui instruções de acesso à memória, como `lw` e `sw`. Durante o desafio, quando uma instrução de leitura da memória for encontrada, o jogador deve abrir o baú com o número hexadecimal correspondente ao endereço de memória lido. De forma semelhante, quando uma instrução de escrita na memória for encontrada, o jogador deve guardar o conteúdo no baú com o número hexadecimal correspondente ao endereço de memória indicado para escrita. É necessário interpretar corretamente as instruções e realizar as ações correspondentes nos baús para completar o desafio com sucesso.



(a) Efeito ao falhar no desafio.



(b) Efeito ao abrir corretamente o baú.

Figura 3.52: Efeitos do Desafio de Acesso a Memória.

Fonte: Próprio autor.

Caso o jogador abra corretamente um baú, um efeito de fogo é criado em volta do jogador, sinalizando que o que estava dentro do baú foi coletado, conforme a Figura 3.52b. Caso o jogador abra incorretamente um baú, um efeito de veneno surge próximo ao jogador, sinalizando sua morte, como mostra a Figura 3.52a.

Ao concluir o desafio, o Viajante realiza uma animação de movimentação, se deslocando para a direita do cenário, e aguarda o jogador próximo a um portal de saltos desativado. Ao interagir, o Viajante explica que o portal será ativado quando o jogador conseguir abrir os dois baús que estão posicionados logo acima, no piso elevado à direita.

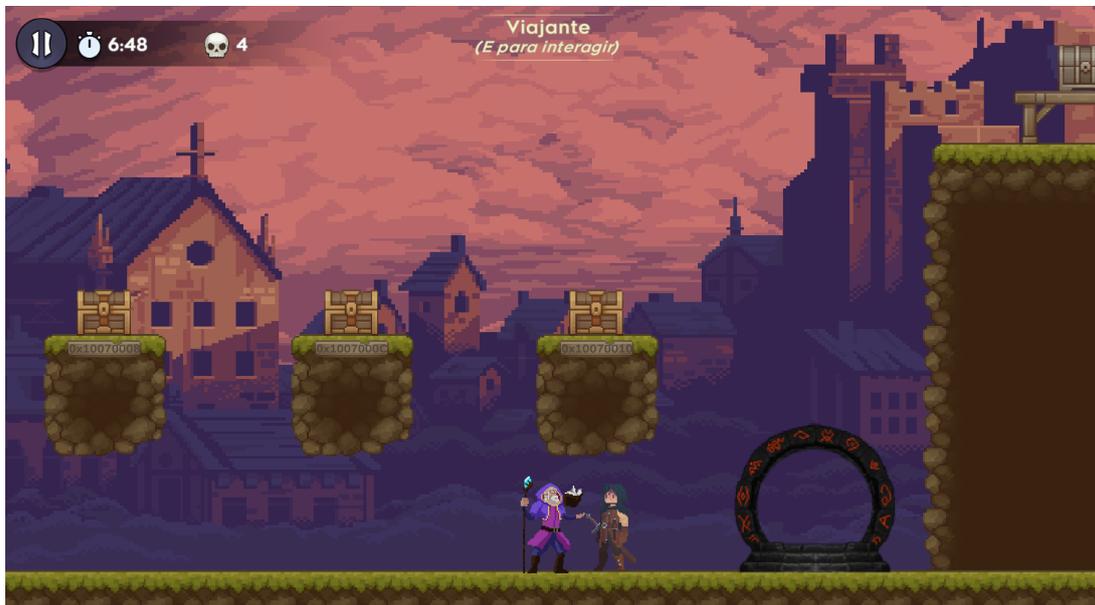


Figura 3.53: Segundo Portal de Saltos do minigame.

Fonte: Próprio autor.

O Viajante explica que para abrir os dois baús, o jogador precisará analisar atentamente o comportamento do código apresentado e, em seguida, clicar nos botões represen-

tando a sequência de execução correta. Os baús são destravados ao pressionar os botões na sequência correta.

Quando o jogador interage com os baús, a UI do desafio é ativada. Nela é apresentado duas listagens contendo as instruções *Assembly*, juntamente com indicações dos endereços iniciais e finais de cada listagem, e dez botões com os endereços de memória.



Figura 3.54: UI do desafio sobre Memória de Código.

Fonte: Próprio autor.

O jogador deve analisar cuidadosamente cada instrução *Assembly* e clicar nos botões de endereço de memória corretos, seguindo a ordem de execução do código. É crucial acertar a sequência de cliques para avançar corretamente. Caso o jogador clique de forma incorreta, o código *Assembly* é alterado, exigindo uma nova análise e tentativa. Porém, se o jogador conseguir clicar corretamente em toda a sequência de execução, o baú é aberto. Ao abrir os dois baús, o portal de saltos é ativado.

Entrando no portal de salto, o jogador é teletransportado para o Vale dos Saltos pela segunda vez, como demonstrado na Figura 3.49. Concluindo esse desafio, o jogador então é transportado para uma nova região, onde o Viajante o aguarda para um novo desafio, o *Desafio da Comida*.



Figura 3.55: Cenário próximo ao Desafio da Comida.

Fonte: Próprio autor.

Na nova região, o jogador encontra o Viajante que o introduz ao Desafio da Comida, conforme a Figura 3.55. O Viajante deseja preparar suas receitas mas elas estão escritas em *Python*. O Personagem não jogável (do inglês, *Non Playable Character*) (NPC) deseja saber o passo-a-passo da receita e os ingredientes necessários, porém o mesmo só entende *Assembly* e precisa que o jogador converta a receita devidamente. Dessa forma, o jogador deve ajudar o Viajante a realizar essa conversão.

O Desafio da Comida consiste em converter o código *Python* em código *Assembly*. Para isso, é apresentado na tela um trecho de código em *Python* e outro em *Assembly*, conforme a Figura 3.56.

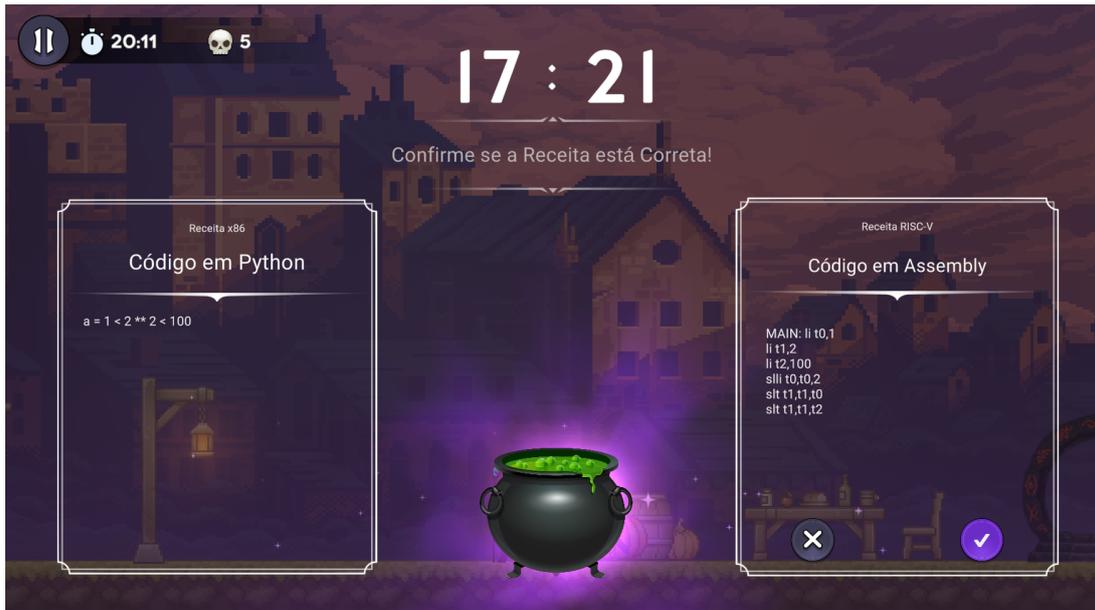


Figura 3.56: Interface do Desafio da Comida.

Fonte: Próprio autor.

O jogador deve fazer uma análise cuidadosa, levando em conta as instruções e a lógica do código em *Python*, e compará-lo com o código em *Assembly* apresentado. O jogador deve selecionar a opção cujo o código *Assembly* tenha o comportamento equivalente do código *Python*. Esse desafio permite que o jogador aplique seus conhecimentos em conversão de código.

Ao concluir o Desafio da Comida, é iniciada uma animação na qual o Viajante caminha para a direita do cenário e é revelado o terceiro portal de saltos, agora ativo. Em sequência, o Viajante explica ao jogador como funciona a dinâmica do último desafio desse minigame que estará disponível após o estudante concluir o Desafio dos Portais pela terceira vez, conforme visto na Figura 3.49.

Ao entrar no portal de saltos e concluir o Desafio dos Portais pela terceira vez, o jogador é transportado para a última região do minigame onde se encontra o desafio final, o *Desafio da Memória de Dados*.



Figura 3.57: Cenário do Desafio de Memória de Dados.

Fonte: Próprio autor.

No Desafio de Memória de Dados, ilustrado na Figura 3.57, o jogador se depara com a tarefa de abrir quatro baús, utilizando suas habilidades de análise de código *Assembly* para determinar os valores armazenados em endereços específicos da memória de dados ao final da execução.

Ao interagir com cada baú, como é mostrado na Figura 3.58, a tela exibe uma representação gráfica da memória de dados à esquerda, o código *Assembly* correspondente à direita e oito blocos de números hexadecimais na região inferior central da tela, divididos em grupos de dois, onde cada grupo representa um byte. Acima desses blocos, é exibido um texto indicando o endereço de memória cujo o valor deve ser inserido pelo jogador utilizando os blocos de números.



Figura 3.58: Desafio de Memória de Dados.

Fonte: Próprio autor.

Ao especificar corretamente os valores para os quatro baús, o Desafio de Memória de Dados é concluído. Nesse momento, o minigame é finalizado e o jogador recebe um *feedback* sobre seu desempenho, conforme a Figura 3.5.

3.2.6 Minigame 2.4

O minigame 2.4 promove a compreensão de códigos em *Assembly RISC-V* por meio de exercícios de construção (em Inglês, *assembly*⁵) e desconstrução (em Inglês, *disassembly*⁶) desses códigos. Ao final do minigame, o aluno irá possuir um melhor entendimento sobre a ISA RISC-V e os tipos de instruções que a compõem através da experiência adquirida nos desafios propostos.

Esse minigame é um dos maiores em relação ao tamanho de mapa, sendo constituído de 3 andares, ou pisos. A medida que o jogador progride, o próximo piso a cima é liberado.

⁵Compilar os mnemônicos do código assembly para o seu código binário equivalente.

⁶Desmontar o código binário para sua representação em mnemônicos.



Figura 3.59: Donzela introduzindo o minigame.

Fonte: Próprio autor.

O minigame começa com o NPC *Donzela* explicando de forma breve os conceitos relacionados ao nível e quais conhecimentos são necessários para o aluno obter êxito nos desafios, conforme a Figura 3.59.

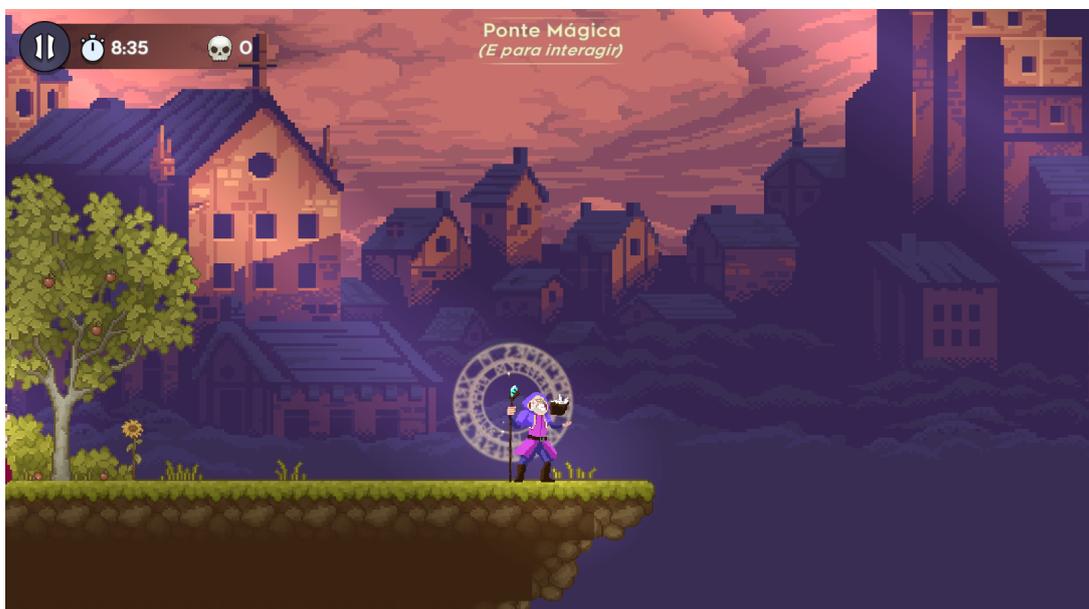


Figura 3.60: Desfiladeiro com o primeiro desafio.

Fonte: Próprio autor.

Ao caminhar para a direita, o jogador irá se deparar com um desfiladeiro e um círculo mágico. Para poder atravessar o desfiladeiro o jogador precisa responder o desafio corretamente, apresentado na Figura 3.60. O desafio em questão consiste em definir os

32 bits do binário equivalente à instrução sendo apresentada na interface. Para facilitar, chamaremos os puzzles como esse, de codificação de instrução, de *AssemblyPuzzle*.

Ao interagir com o círculo mágico, é aberta uma Interface de Usuário (do inglês, *User Interface*) (UI) onde o jogador define os bits da instrução apresentada em binário, conforme a Figura 3.61. Para mudar o valor de cada bit basta clicar no botão correspondente, cada bit varia ciclicamente entre 0 e 1.



Figura 3.61: UI do puzzle de codificação de instrução.

Fonte: Próprio autor.

Ao definir os bits da instrução corretamente o jogador deve confirmar a resposta pressionando no botão “Confirmar”, conforme a Figura 3.62a. Se a resposta estiver correta, uma ponte irá surgir, permitindo que o jogador atravesse o desfiladeiro, conforme a Figura 3.62b.



(a) UI antes de confirmar a resposta.



(b) Ao acertar, a ponte surge.

Figura 3.62: Jogador inserindo a resposta no *AssemblyPuzzle*.

Fonte: Próprio autor.

Seguindo a diante, o jogador se deparará com outro desfiladeiro, onde ele terá que completar outro *AssemblyPuzzle*, conforme descrito acima. Existem vários *AssemblyPuzzles* espalhados pelo mapa do minigame, cada vez que um é acionado, a pergunta a ser apresentada é consumida de um banco de dados de questões⁷.

Após atravessar 2 desfiladeiros, o jogador encontra o NPC *Fanstasma*, conforme a Figura 3.63. Ao interagir com ele, o jogador adquire informações sobre um novo tipo de desafio, onde o aluno precisará desmontar códigos binários para suas instruções *Assembly* em ISA RISC-V equivalente. Esses puzzles são chamados de *DisassemblyPuzzle*.

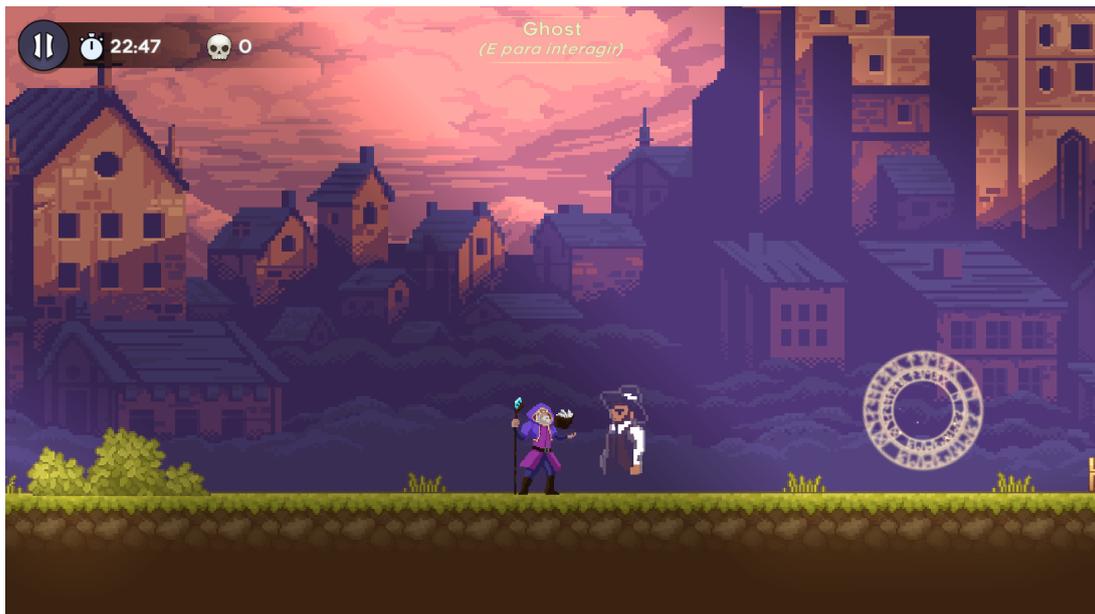


Figura 3.63: NPC *Fantasma* à direita do jogador.

Fonte: Próprio autor.

À direita do Fantasma, o jogador encontra um círculo mágico interagirável, ao ativá-lo é mostrado um painel no canto inferior esquerdo da tela, conforme a Figura 3.64. O painel mostra um código binário, uma sequência de 32 bits, e um campo onde o jogador deve inserir a instrução RISC-V equivalente.

⁷As questões são pré-cadastradas nesse banco de dados, e é possível buscá-las a qualquer momento de execução do jogo. Esse tema é discutido com profundidade na seção 3.3.

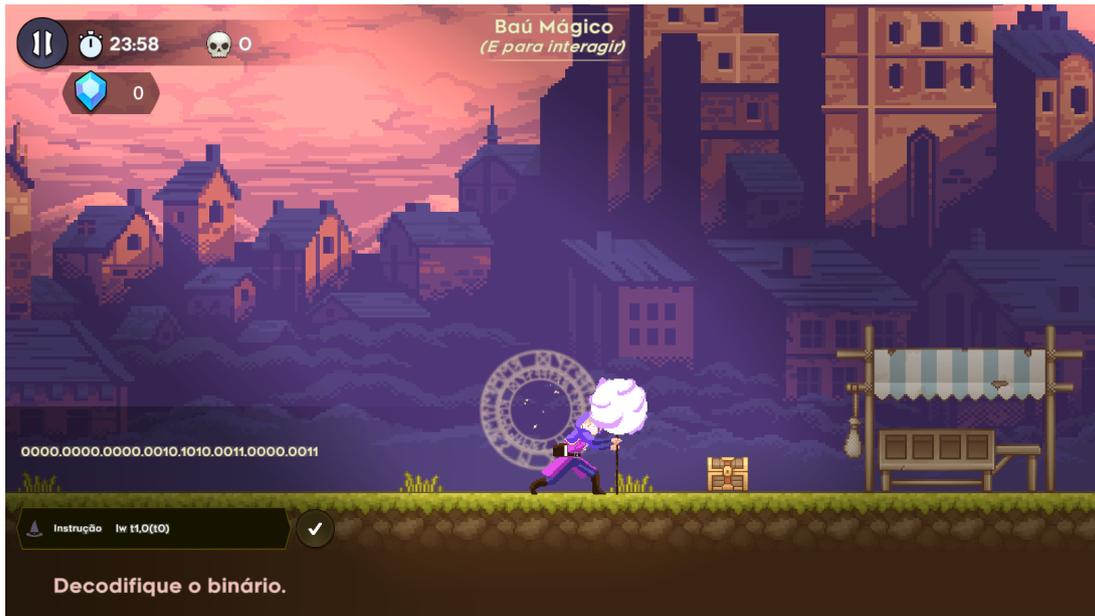


Figura 3.64: Jogador inserindo resposta no painel.

Fonte: Próprio autor.

Inserindo a instrução corretamente, uma interação audiovisual acontece, onde o baú se abre e o jogador é recompensado com uma gema, conforme as Figuras 3.64 a 3.65.

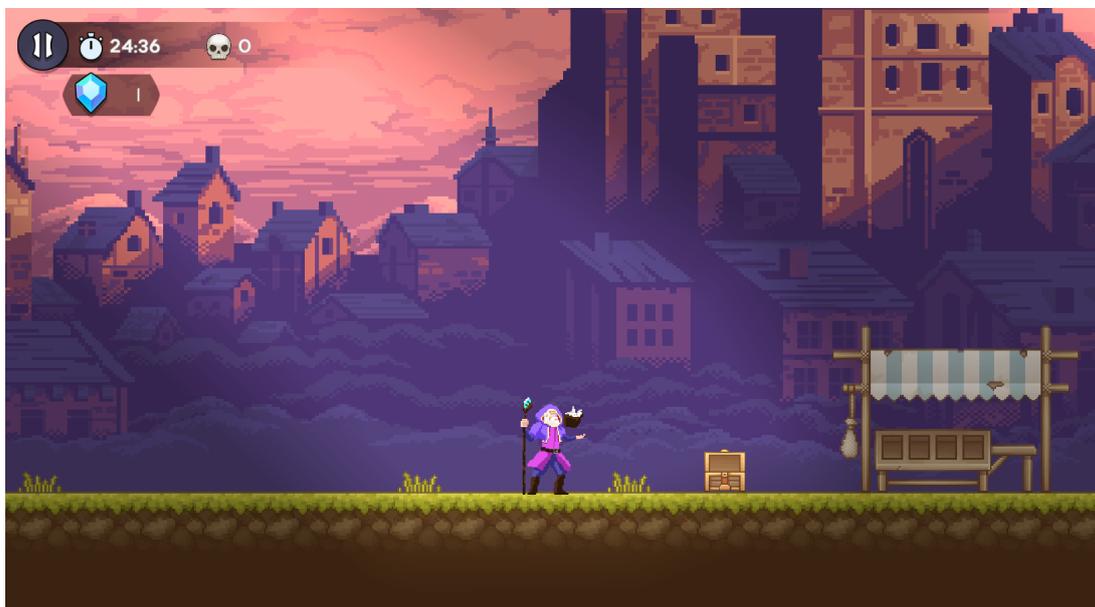


Figura 3.65: Ao acertar, o baú se abre e 1 gema é adquirida.

Fonte: Próprio autor.

As gemas compõem uma mecânica única desse minigame e o jogador precisa coletá-las para concluir o nível. Como na demonstração do *DisassemblyPuzzle* acima, as gemas são

adquiridas ao abrir baús espalhados pelo mapa. O jogador encontra os baús seguindo o percurso planejado do mapa. A ideia em torno dessa mecânica é aumentar a imersão do jogador com o mundo a medida que promove novas interações. Aqui é aplicado o conceito de *missões*, um elemento chave de jogos e frequentemente utilizado na gamificação [1].

Em frente, o jogador se depara com um vilarejo abandonado, conforme a Figura 3.66. O diálogo com o NPC Fantasma explica a situação dizendo que o local foi abandonado após ladrões saquearem o vilarejo. Mas, se o jogador for capaz de encontrar todas as gemas roubadas, os moradores voltarão para a vila. Essa trama faz parte da missão principal do nível e utiliza a mecânica das gemas, como explicado anteriormente.

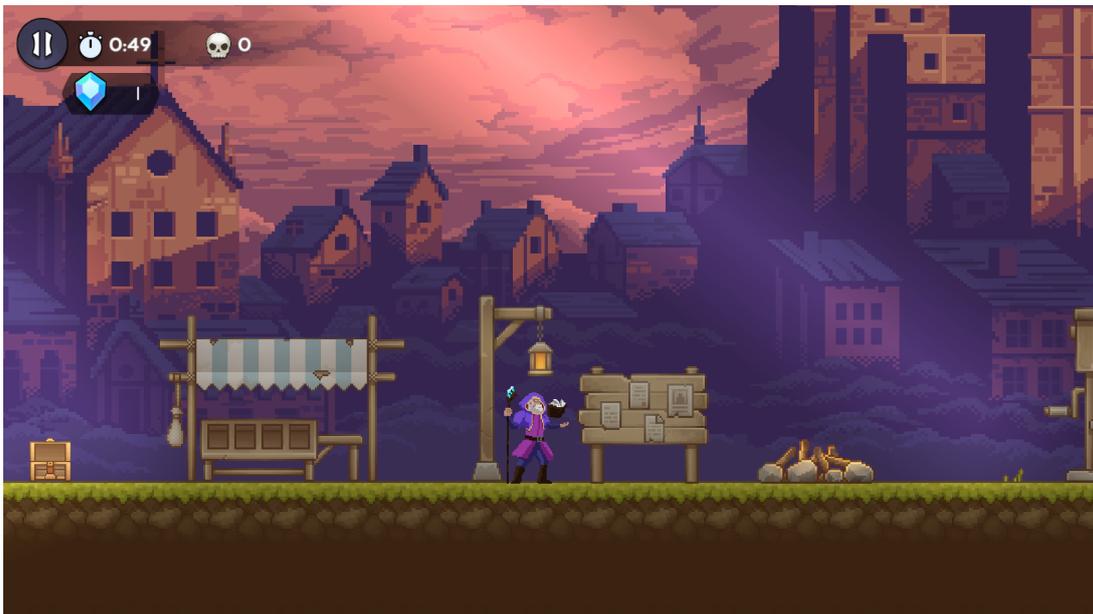


Figura 3.66: Vilarejo abandonado.

Fonte: Próprio autor.

Ao final do primeiro piso, o jogador encontra um portal desativado, como visto na Figura 3.67. Para ativá-lo o jogador precisa passar pelo desafio apresentado. No caso dos portais, são aplicados os *AssemblyPuzzles* e apresentam a mesma dinâmica do desafio inicial no desfiladeiro, conforme a Figura 3.61.

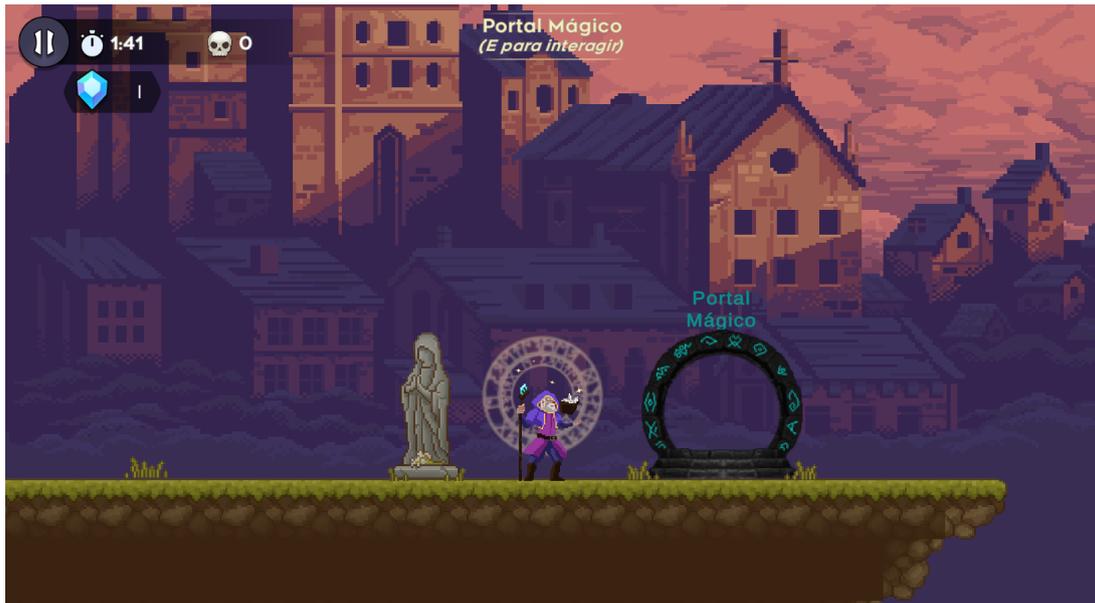


Figura 3.67: Vilarejo abandonado.

Fonte: Próprio autor.

Para completar o desafio e ativar o portal, o estudante deve codificar a instrução apresentada, como apresentado na Figura 3.68. Feito isso, o portal é ativado e o jogador pode seguir para o próximo andar.

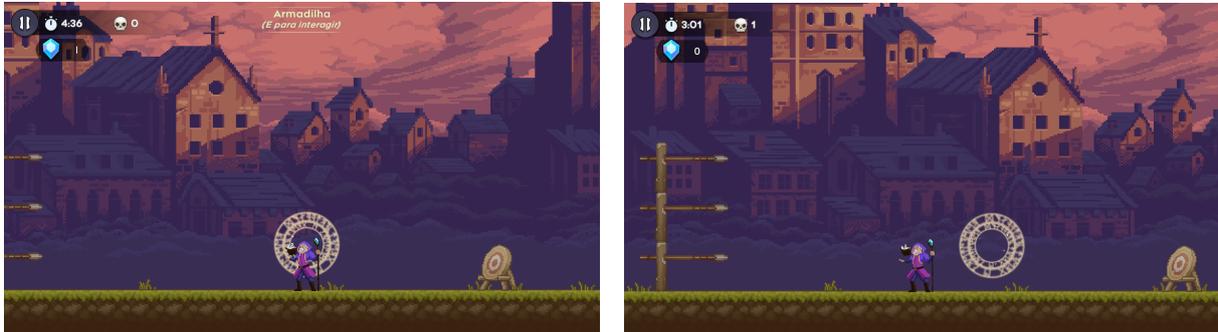


Figura 3.68: *DisassemblyPuzzle* do portal.

Fonte: Próprio autor.

No segundo piso, é apresentado uma nova interação. No canto esquerdo da tela há lanças que impedem o jogador de prosseguir, conforme a Figura 3.69b. Caso o jogador se

aproxime, a armadilha é acionada e as lanças são disparadas em direção ao jogador. Dessa forma, o estudante morre e renasce no último *checkpoint*⁸. Para desativar a armadilha, o jogador precisa completar o desafio interagindo com o círculo mágico. O desafio é um *DisassemblyPuzzle* e segue a mesma dinâmica explicada anteriormente.



(a) Círculo mágico para desativar Armadilha. (b) Lanças impedem o jogador de prosseguir.

Figura 3.69: *AssemblyPuzzle* das lanças.

Fonte: Próprio autor.

A partir desse desse ponto, as mecânicas e interações que o aluno encontrará são repetições do que foi apresentado até agora. No entanto, a medida que o jogador progride, os desafios se tornam mais difíceis. Em outras palavras, no primeiro piso do minigame, os puzzles são fáceis e estimulam o aluno às primeiras interações com o ambiente. Ao avançar, os puzzles se tornam mais difíceis para promover um cenário desafiador e manter o estudante engajado [1]. Dessa forma, os desafios do terceiro piso são os mais difíceis.

Em certo ponto do terceiro piso, o jogador se depara com uma variação da UI, como mostra a Figura 3.70 à seguir.

⁸Termo utilizado em jogos para indicar um local (ou estado do jogo) seguro, e que foi visitado pelo jogador previamente.



Figura 3.70: *AssemblyPuzzle* pedindo codificação em hexadecimal.

Fonte: Próprio autor.

Conforme a Figura 3.70 mostra, é pedido ao jogador a codificação da instrução apresentada em hexadecimal. Além de aumentar a dificuldade do desafio, essa variação na escrita apresenta uma alternativa à representação binária para notação dos bytes da instrução: 2 dígitos hexadecimais representam 1 byte, logo, uma instrução de 4 bytes (32 bits) é representável por 8 dígitos hexadecimais. Por ser mais compacta, a notação em hexadecimal é comumente utilizada na computação.

Ao final do terceiro piso, após coletar a terceira e última gema, o jogador encontra o Viajante novamente, conforme a Figura 3.71. O Viajante parabeniza o estudante por recuperar as gemas dos aldeões e o instrui a seguir em frente e pegar o portal no final do piso para retornar ao vilarejo.



Figura 3.71: *AssemblyPuzzle* pedindo codificação em hexadecimal.

Fonte: Próprio autor.

No vilarejo, agora populado por seus moradores, o jogador interage com o NPC *Donzela* quem o parabeniza por completar todos os desafios, conforme a Figura 3.72.



(a) Vila com seus moradores.



(b) Jogador conversando com a *Donzela*.

Figura 3.72: Vila restaurada e tela de conclusão.

Fonte: Próprio autor.

Em seguida, é apresentada a tela de conclusão do nível mostrando o desempenho do estudante e um resumo dos tópicos exercitados no minigame, conforme a Figura 3.5.

3.3 Detalhes de Implementação

Os minigames estão hospedados no Moodle [33] Aprender da UnB. A plataforma oferece uma variedade de ferramentas que dão suporte ao aprendizado [34], como fóruns de dú-

vidas, questionários, compartilhamento de arquivos (como slides, vídeos e códigos), entre outras.

A plataforma Moodle oferece suporte ao SCORM versão **1.2** [10]. Como explicado na Seção 2.3, o SCORM consiste em um conjunto de padrões que define como deve ser a interação entre o LMS e o pacote SCORM. A tecnologia SCORM é essencial para o projeto, pois permite a submissão de um código executável externo, independente da plataforma, totalmente encapsulado.

Além disso, também possibilita a troca de informações entre o pacote SCORM e a plataforma, como o acesso do aluno à tarefa, progresso realizado, nota obtida na tarefa, tempo utilizado, entre outros.

A interação do jogador com os conteúdos de ISC nos minigames ocorre principalmente por meio dos desafios. Esses são alimentados por um Sistema de Diálogo, que armazena os textos das questões, suas respostas e todos os diálogos do jogo. Cada minigame, possui um banco de dados de diálogos específico, e cada personagem falante no jogo tem um ou mais diálogos armazenados nesse banco de dados.

Os bancos de dados podem ser carregados a partir de arquivos externos no formato *.xml*, o que permite a fácil edição de diálogos existentes ou que novos diálogos sejam adicionados pelo docente. Essa funcionalidade é especialmente útil para expandir o banco de dados de questões disponíveis nos desafios, sem a necessidade de alterar ou recompilar o código-fonte do sistema.

Na Figura 3.73 são apresentados os principais módulos que compõem o Sistema de Diálogo. Esse sistema é baseado no *plugin Dialogue System for Unity*, produzido pela empresa *Pixel Crushers* [35]. O *plugin* oferece todas as funcionalidades necessárias para o gerenciamento dos diálogos, incluindo a exibição de balões de diálogo na tela, a captura das respostas do usuário, a aplicação de condições ou eventos, o carregamento de bancos de dados a partir de arquivos externos, entre outros recursos.

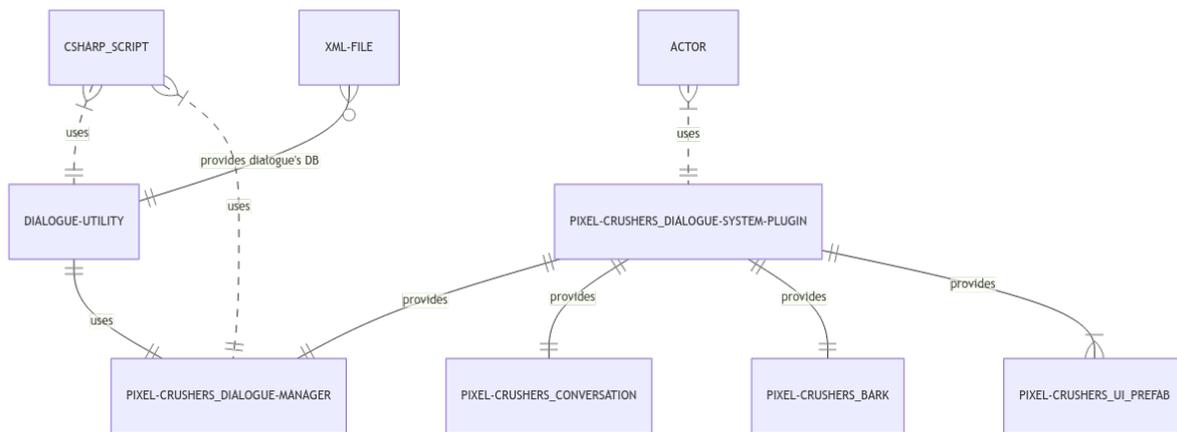


Figura 3.73: Diagrama de relações entre os componentes do Sistema de Diálogo.

Fonte: Próprio autor.

Visualizando o diagrama de cima para baixo temos:

- **CSHARP_SCRIPT**: Script C#, representa *scripts* que o desenvolvedor pode criar para interagir com o sistema.
- **XML-FILE**: Arquivo *.xml*, arquivo gerado ao exportar diálogo através da aplicação *TalkerMaker Deluxe* [36].
- **DIALOGUE-UTILITY**: Sistema desenvolvido para carregar arquivos *.xml* e fazer o *parse* das questões presentes no banco de dados.
- **ACTOR**: Entidade no mundo, pode utilizar os recursos do Sistema de Diálogo, como: **Bark()**, **StartConversation()** e utilizar UI de diálogos.

Utilizando as funcionalidades fornecidas pelo Sistema de Diálogo, foi desenvolvido um sistema superconjunto, que permite a integração de *questionários* como uma forma de diálogo. Por meio desse sistema, o docente pode cadastrar novas questões e diálogos no MagicBits, seguindo determinadas convenções, sem a necessidade de acessar o código-fonte ou recompilar o projeto.

Para editar os arquivos *.xml* foi utilizado a aplicação *TalkerMaker Deluxe* [36]. Ela é compatível com o plugin *Dialogue System for Unity* utilizado pelo MagicBits e permite a fácil edição dos arquivos sem precisar acessar o código-fonte do projeto pela Unity. A seguir, é apresentado um exemplo de diálogo aberto na ferramenta *TalkerMaker Deluxe*, o qual contém informações de uma questão, conforme a Figura 3.74.

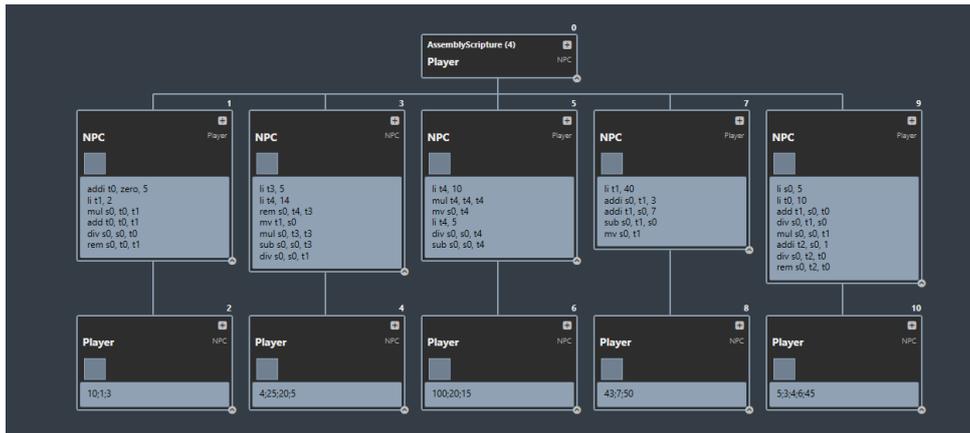


Figura 3.74: Exemplo de diálogo aberto no *TalkerMaker Deluxe*.

Fonte: Próprio autor.

Para carregar os arquivos *.xml* a seguinte convenção de diretório é seguida:

```
"./Resources/{s_minigameName}_{m_levelName}_DB.xml"
```

Onde, cada variável entre “” representa:

- *s_minigameName*: variável representando o nome do minigame, escrito na forma *Minigame_<Módulo>_<Y>*, onde *Módulo* é o módulo cujo o minigame pertence, *Y* um sufixo qualquer (geralmente com valor “x”) para agrupar minigames do mesmo módulo. Exemplos de nomeação válida: **Minigame_2_x**, **Minigame_1_x**.
- *s_levelName*: variável representando o nome do nível (e.g., representa o arquivo do minigame em si). Escrito no formato *Level<ID>*, onde *ID* é a identificação única do nível daquele módulo. Exemplos de valores válidos: **Level1**, **Level2**, **Level3**.

Ao se referir ao minigame 2.3, *s_minigameName* seria **Minigame_2_x** e *Level<ID>* seria **Level3**. Dessa forma, o caminho do arquivo do Banco de Dados dos diálogos para esse minigame seria:

```
"./Resources/{Minigame_2_x}_{Level3}_DB.xml"
```

Seguindo o exemplo acima, o docente criaria o arquivo **Minigame_2_x_Level3_DB.xml** utilizando a ferramenta *TalkerMaker Deluxe* e o salvaria no diretório *./Resources/*. Feito isso, as rotinas do MagicBits vão, em tempo de execução, carregar as informações do arquivo *.xml* e atualizar os diálogos.

O guia atualizado sobre a adição de diálogos pode ser consultado neste repositório do GitHub⁹.

⁹<https://github.com/edassis/DialogueDBs>

Capítulo 4

Resultados Obtidos

A disciplina de ISC foi ministrada por um único professor nos semestres de 2021.2 a 2023.1. Durante esse período, as questões das provas foram elaboradas com dificuldades similares, muitas vezes mantendo a ideia principal e realizando a randomização de valores. Após a conclusão da primeira e segunda prova do semestre de 2023.1, foram coletadas as notas de cada aluno para análise e comparação com os três semestres anteriores.

4.1 Análise das notas com semestres anteriores

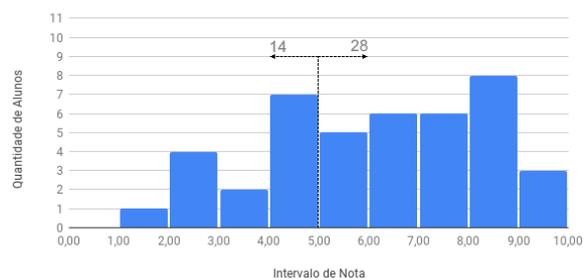
Para cada semestre, foi criado um histograma das notas onde o eixo horizontal representa o intervalo de notas e o eixo vertical a quantidade de alunos. Para essa análise, os alunos que não fizeram uma das duas primeiras provas foram excluídos dos cálculos afim de tratar os dados. Para os alunos considerados, foram calculados a média aritmética e o desvio padrão das notas da primeira (P1) e da segunda prova (P2), bem como o número de alunos que obtiveram uma nota igual ou superior a 5,00 pontos¹e menor a 5,00.

Na Figura 4.1 e Figura 4.2, é possível observar um impacto positivo no semestre de 2023.1, no qual os minigames foram utilizados, em comparação com os semestres anteriores nos quais os minigames não foram adotados.

¹Critério para o aluno obter aprovação em qualquer disciplina da UnB.

Histograma Prova 1 - 2023/1

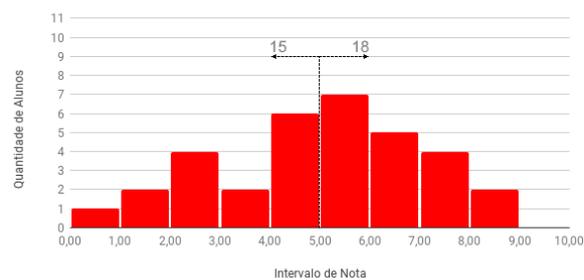
Intervalo de 1 ponto



(a) Histograma P1 2023.1.

Histograma Prova 1 - 2022/2

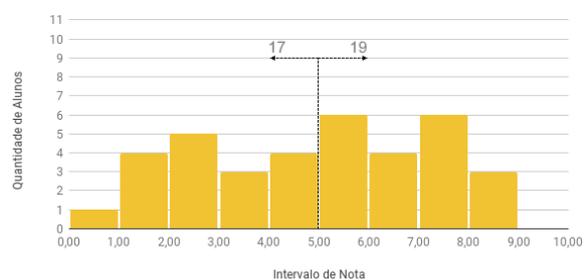
Intervalo de 1 ponto



(b) Histograma P1 2022.2.

Histograma Prova 1 - 2022/1

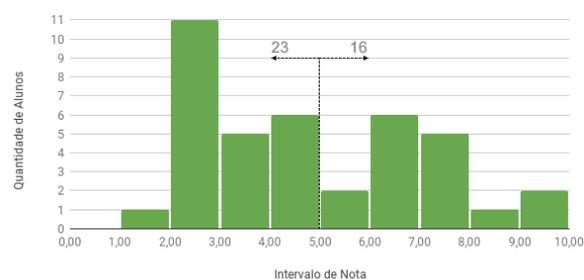
Intervalo de 1 ponto



(c) Histograma P1 2022.1.

Histograma Prova 1 - 2021/2

Intervalo de 1 ponto

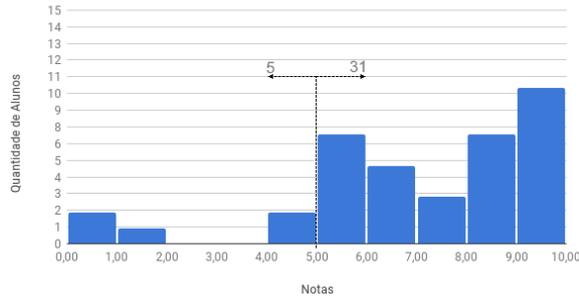


(d) Histograma P1 2021.2.

Figura 4.1: Histogramas da primeira prova.

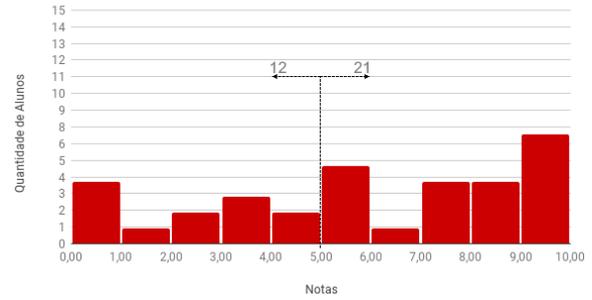
Fonte: Próprio autor.

Histograma Prova 2 - 2023/1



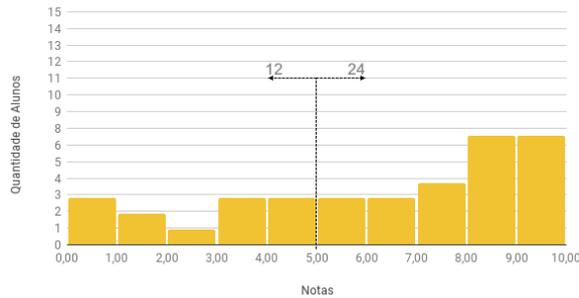
(a) Histograma segunda prova 2023.1.

Histograma Prova 2 - 2022/2



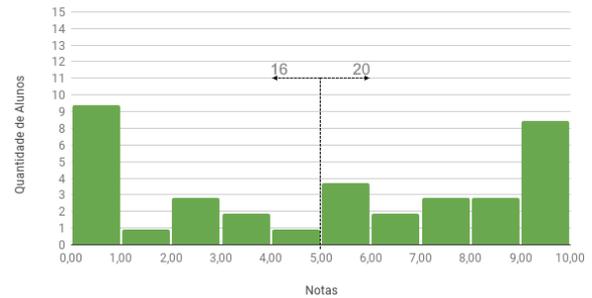
(b) Histograma segunda prova 2022.2.

Histograma Prova 2 - 2022/1



(c) Histograma segunda prova 2022.1.

Histograma Prova 2 - 2021/2



(d) Histograma segunda prova 2021.2.

Figura 4.2: Histogramas da segunda prova.

Fonte: Próprio autor.

Podemos observar que no semestre 2023.1 houve uma maior quantidade de alunos ocupando a região central do gráfico, com uma distribuição mais uniforme. Na Tabela 4.1 a seguir, é apresentado de forma textual e detalhada mais informações sobre as médias.

Tabela 4.1: Resultados das provas dos semestres entre 2023.1 e 2021.2.

Semestre	Prova 1 (P1)			Prova 2 (P2)			$\Delta\mu$ (P1+P2)
	μ	σ	% Ap.	μ	σ	% Ap.	
2023.1	6,08	2,22	66%	7,04	2,58	86%	0,00 %
2022.2	4,98	1,95	54%	5,82	3,17	63%	+21,48 %
2022.1	4,97	2,43	54%	6,06	3,06	66%	+18,95 %
2021.2	4,81	2,18	41%	5,02	3,68	55%	+33,47 %

A Tabela 4.1 apresenta as médias (μ) e desvios padrões(σ) das provas 1 (P1) e provas 2 (P2) ao longo dos semestres. A última coluna mostra o desempenho (Δ) percentual do semestre 2023.1 em relação aos demais semestres, computado utilizando a média aritmética das provas ($P1 + P2$).

No semestre de 2023.1, no qual os minigames foram aplicados, a turma obteve uma média de 6,08 pontos na P1. Dos 42 alunos que realizaram a prova, 28 alcançaram uma nota igual ou superior a 5,00 pontos, representando 66% dos alunos da turma. Nos semestres de 2022.2, 2022.1 e 2021.2 a proporção de alunos que alcançaram uma nota acima de 5,00 pontos foi de 54%, 54% e 41% respectivamente.

Quanto à P2, a média dos alunos no semestre 2023.1 foi de 7,04 pontos, e a proporção de alunos que obtiveram uma nota igual ou superior a 5,00 pontos foi de 86%, com um total de 37 alunos que realizaram a prova. Nos semestres 2022.2, 2022.1 e 2021.2 a proporção de alunos que tiraram nota igual ou superior a 5,00 pontos foi de 63%, 66% e 55% respectivamente.

Ao comparar a média aritmética das notas da primeira e segunda prova somadas do semestre 2023.1 com os demais semestres, última coluna da Tabela 4.1, temos um aumento positivo variando entre 18,95% (2022.1) e 33,47% (2021.2).

4.2 Impacto dos minigames nas provas

Uma análise mais abrangente foi conduzida para investigar o impacto dos minigames no desempenho dos alunos. Onde, foi feita a correlação entre a pontuação dos alunos nos minigames com suas respectivas notas das provas dos módulos 1 e 2, nos quais os minigames foram aplicados.

Inicialmente, foi examinado o número de alunos que abriram ou não os minigames em cada módulo da disciplina. Os dados correspondentes podem ser observados na Tabela 4.2.

Tabela 4.2: Quantidade de alunos que abriram ou não abriram os minigames

Minigame	Abriram	Não abriram
1.1	40	3
1.2	37	6
2.1	23	20
2.2	24	19
2.3	16	27
2.4	20	23

No primeiro módulo é evidenciado que a grande maioria dos alunos acessaram os minigames, onde 93,02% dos alunos abriram o minigame 1.1 (i.e., 40 alunos abriram e 3 alunos não abriram), e 86,05% abriram o minigame 1.2 (i.e., 37 alunos abriram e 6 não abriram). Ao analisar o segundo módulo, nota-se que muitos alunos deixaram de acessar os minigames, onde 53,49% dos alunos acessaram o minigame 2.1, 55,81% dos alunos

acessaram o minigame 2.2, 37,21% dos alunos acessaram o minigame 2.3, e 46,51% dos alunos acessaram o minigame 2.4.

Os dados referentes ao segundo módulo são alarmantes, pois dão indícios no aumento de alunos desinteressados em participar dos minigames. Isso pode ser atribuído à dificuldade dos desafios ou por considerarem que os minigames não são uma boa opção para a prática do conteúdo da disciplina. Outro ponto que fundamenta essas suspeitas é a opinião dos alunos no formulário de avaliação dos minigames do módulo 2, semestre 2023.1, onde o minigame 2.4 foi o mais criticado pelos alunos, com a maior quantidade de votações na pergunta “Qual minigame você menos gostou?”, conforme mostra o Anexo B.

Considerando os alunos que obtiveram um bom desempenho nas provas (i.e., média maior ou igual a 5), foi feita uma análise separando os alunos que **jogaram** os minigames dos que **não jogaram**, conforme a Tabela 4.3. Essa comparação tem o objetivo de identificar se os estudantes que jogaram os minigames alcançaram uma boa nota.

Tabela 4.3: Quantidade de alunos com bom desempenho.

Prova	Nota ≥ 7		Nota ≥ 5	
	Jogaram	Não Jogaram	Jogaram	Não Jogaram
1	17	0	27	1
2	17	3	22	10

Ao analisar os alunos que **abriram e jogaram** os minigames do módulo 1, 27 alunos obtiveram nota maior ou igual a 5,00 pontos, e desses alunos, 17 obtiveram nota maior ou igual a 7,00 pontos. No módulo 2, 22 alunos obtiveram nota maior ou igual a 5,00 pontos, e desses alunos, 17 obtiveram uma nota maior ou igual a 7,00 pontos.

Por outro lado, analisando os alunos que **não jogaram** os minigames, apenas 1 aluno **não** participou de nenhum dos minigames do módulo 1 e ainda assim conseguiu obter uma nota igual ou superior a 5,00 pontos. No módulo 2, foram 10 alunos que **alcançaram** uma nota igual ou superior a 5,00 pontos e **não** jogaram nenhum dos minigames, e desses alunos, apenas 3 obtiveram uma nota igual ou superior a 7,00 pontos.

De forma semelhante, também foram analisados os dados referentes aos alunos com baixo desempenho nas provas (i.e., média inferior a 5) e sua participação nos minigames, conforme a Tabela 4.4.

Tabela 4.4: Quantidade de alunos com baixo desempenho.

Prova	Nota < 5	
	Jogaram	Não Jogaram
1	11	3
2	1	5

Analisando a Tabela 4.4, é observado que 14 alunos obtiveram uma nota inferior a 5,00 pontos na primeira prova, e desses alunos, 11 **jogaram** os minigames e 3 **não jogaram**. Em relação à segunda prova, é observado que 6 alunos tiveram nota inferior a 5,00 pontos, e desses alunos, apenas 1 **jogou** os minigames e 5 **não jogaram**.

Com base nessas observações e os dados apresentados nas Tabelas 4.3 a 4.4, é possível observar uma ligação entre o aluno jogar os minigames e obter um bom desempenho nas provas.

4.3 Correlação entre prova e minigames

Para melhor entender a correlação entre “o aluno jogar os minigames” e “o aluno ser aprovado”, foram construídos dois gráficos de dispersão. A Figura 4.3 apresenta essa correlação para o primeiro módulo e a Figura 4.4 para o segundo módulo. Nos dois gráficos, o eixo x representa os alunos, enquanto o eixo y representa as notas alcançadas. Como as notas dos alunos variam de 0 à 10 e a dos minigames de 0 à 100, as notas dos minigames foram divididas por 10 afim de corrigir os intervalos.

Gráfico de dispersão - Módulo 1

Entre prova e minigames

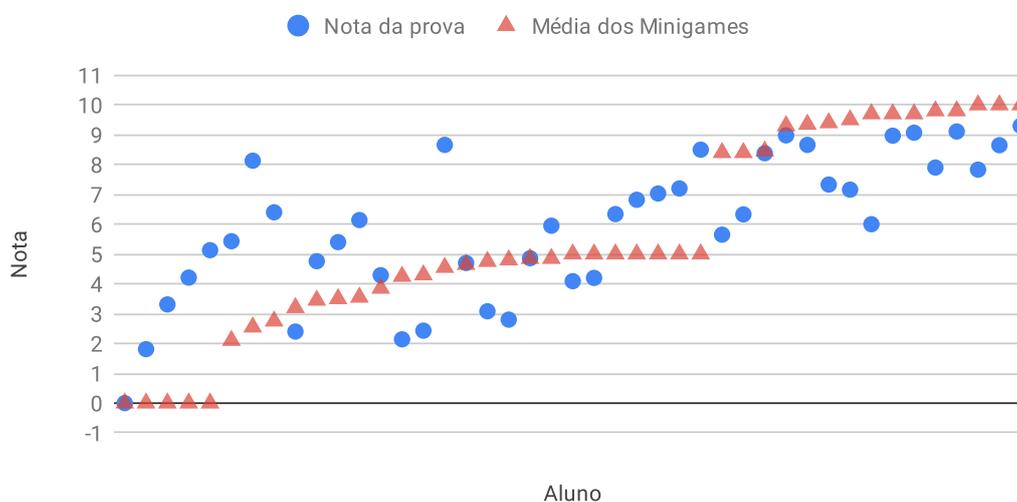


Figura 4.3: Relação entre notas do primeiro módulo.

Fonte: Próprio autor.

Conforme mostra a Figura 4.3, os círculos azuis representam as notas da prova e os triângulos vermelhos representam as médias nos minigames. É possível observar nesta figura uma proximidade entre a dispersão dos círculos azuis e dos triângulos vermelhos. À medida que a média dos minigames sobe a nota das provas também sobe. O contrário também é verdade, a medida que a nota dos minigames desce a nota das provas também desce.

No segundo módulo, foi observado que vários alunos obtiveram pontuação zero nos minigames mas ainda assim conseguiram notas elevadas nas provas, conforme mostra a região esquerda da Figura 4.4. Por outro lado, todos os alunos que alcançaram uma média igual ou superior a 5,00 pontos nos minigames também obtiveram notas iguais ou superiores a 5,00 pontos na prova, fato que também é verdade na Figura 4.3.

Gráfico de dispersão - Módulo 2

Entre prova e minigames

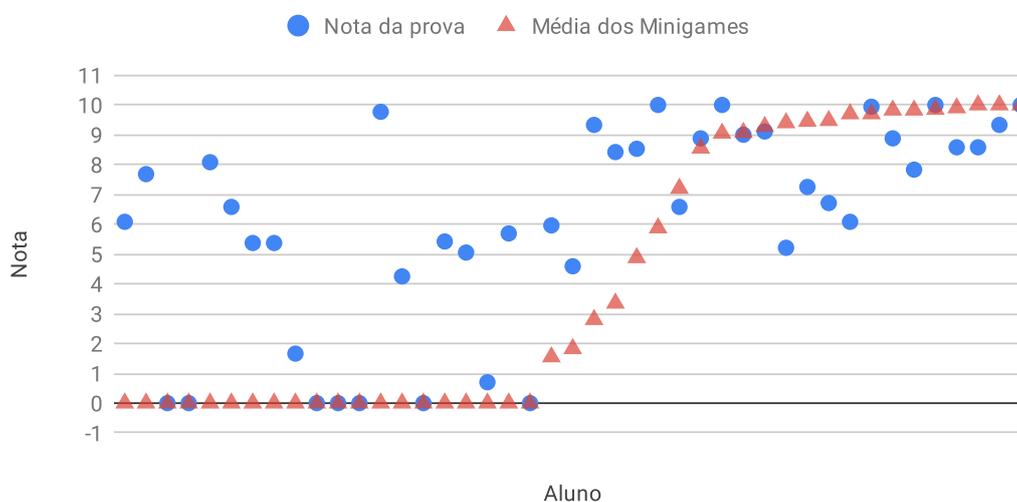


Figura 4.4: Relação entre notas do segundo módulo.

Fonte: Próprio autor.

Esse comportamento, de alunos que não fizeram os minigames obterem um bom desempenho nas provas, reforçam nossas suspeitas da dificuldade excessiva em completar os desafios ou a falta de interesse pela dinâmica proposta.

Ao aplicar a correlação de Pearson [37] entre a média dos minigames e a nota das provas, obtemos um coeficiente de 0,70 para o primeiro módulo e 0,65 para o segundo módulo. Esses valores indicam que há uma relação moderadamente positiva entre os fatores, ou seja, os desempenhos nos minigames não são essenciais para obter uma boa nota, mas contribuem para o desempenho dos estudantes nas provas.

4.4 Opinião dos alunos

Ao final do semestre 2023.1, foi realizado um formulário anônimo de avaliação da disciplina, onde 8 alunos responderam, conforme Anexo B. Essa carência de dados limitou a identificação da opinião média da turma a respeito da disciplina. No entanto, algumas respostas para perguntas específicas indicam que a criação de minigames para o módulo 3 poderia contribuir no aprendizado.

Analisando as respostas dos alunos para a pergunta “Qual conteúdo de ISC você achou mais difícil? Por quê?”, foi observado que os termos “Transistores”, “Circuitos Lógicos” e “Bloco de Controle” são frequentemente utilizados. Esses temas constituem o módulo 3 de ISC, o qual tem um foco grande em *hardware* e nos conceitos de Elétrica e Física

envolvidos. Sendo assim, o módulo 3 tem uma natureza interdisciplinar, o que traz uma complexidade inerente.

Outra resposta relevante menciona explicitamente a falta de minigames para o módulo 3. Na resposta o aluno expressa que, apesar da dificuldade encontrada em resolver os desafios dos minigames do módulo 1 e 2, a presença deles proporcionou uma melhor absorção dos conteúdos. Visto que não foram desenvolvidos minigames para o módulo 3, essa experiência adicional que o aluno pôde adquirir pelo MagicBits não ocorreu nesse módulo. Dessa forma, podemos observar uma necessidade em desenvolver minigames para o módulo 3 em trabalhos futuros.

Capítulo 5

Conclusão

Este trabalho propôs uma metodologia de aprendizagem em que o conteúdo visto em sala de aula é praticado através de um jogo sério, o MagicBits. Esse estudo objetiva aumentar o engajamento dos alunos nas atividades da disciplina.

Com o intuito de proporcionar uma experiência de aprendizado mais dinâmica e envolvente, foram desenvolvidos seis minigames, divididos em dois módulos da disciplina. Para o primeiro módulo, foram criados dois minigames, enquanto o segundo módulo contou com quatro minigames. Cada um desses minigames foi projetado com exercícios interativos, apresentados na forma de desafios, inseridos em um contexto de universo lúdico.

Cada minigame aborda conceitos específicos da disciplina, oferecendo uma variedade de desafios que permitem ao aluno exercitar suas habilidades de resolução de problemas e aprofundar seus conhecimentos nos conteúdos abordados. Além disso, os minigames são projetados para serem facilmente acessíveis e compreensíveis, para que os alunos possam usufruir da experiência de aprendizado de forma descomplicada.

Com base nos dados advindos dos resultados obtidos, foi observado um aumento considerável na média dos alunos de ISC do semestre 2023.1, mesmo se considerarmos a primeira e segunda prova isoladamente, conforme a Tabela 4.1. Isso nos dá indícios que a utilização da gamificação na disciplina ISC tem um impacto positivo, contribuindo para o aumento do engajamento e interação do aluno com a disciplina.

É importante considerarmos o escopo do experimento, cujo consistiu em alunos do Ensino Superior de um curso da área da computação, onde o ambiente laboratorial foi a disciplina ISC, voltada à programação e arquitetura de computadores. Nesse contexto, é plausível dos interesses e predisposições dos alunos ressonarem com os elementos do MagicBits e sua metodologia de ensino. Mas nada garante que o mesmo experimento obteria resultados semelhantes aos apresentados em um contexto diferente, com alunos com perfis diferentes ou disciplinas fora da computação.

Além disso, é incerto na nossa metodologia quais características e componentes da gamificação contribuíram para os nossos resultados. Como os experimentos propostos aplicaram empiricamente uma combinação de diversos elementos de jogos, é difícil identificar quais desses elementos foram mais relevantes, ou, quais agiram como detratores [8]. Além disso, a avaliação de resultados foi feita utilizando scores e formulários, uma abordagem “caixa preta”, onde se é feita uma análise antes e após o experimento.

Apesar do que foi mostrado contribuir para a utilização da gamificação na educação, mais estudos são necessários para identificar, de forma sistemática, quais são os aspectos e fatores chave que a experiência gamificada deve oferecer para aumentar o engajamento dos alunos.

5.1 Trabalhos Futuros

Analisando os formulários anônimos aplicados após o período de experimentação dos minigames, um ponto de bastante crítica pelos alunos foi o alto grau de dificuldade de alguns desafios, fazendo alguns deles desistirem de utilizarem os jogos e estudarem na metodologia tradicional. Esse fato aponta uma deficiência do MagicBits, cujo não personaliza a dificuldade conforme as características de cada aluno, a dificuldade dos desafios é igual para todos.

Assim, é necessário o desenvolvimento de sistemas para balancear os desafios propostos conforme o perfil de cada aluno. Além de um refinamento na dificuldade padrão dos desafios, é importante que o MagicBits contenha heurísticas capazes de ajustarem dinamicamente a dificuldade dos exercícios conforme o desempenho atual do aluno. Ao oferecer uma experiência customizada, a frustração pela falha diminui e contribui para o engajamento [3].

Um tema de bastante crítica na literatura que se aplica ao nosso estudo é a falta de introspecção dos sistemas de gamificação ou jogos educacionais. Apesar de o MagicBits ter obtido êxito em promover o engajamento e contribuir positivamente para o desempenho dos alunos nas provas, não sabemos quais dos elementos de jogos aplicados contribuíram para esse resultado [8]. Desenvolver sistemas de monitoração que gere um histórico das ações tomadas pelo jogador se mostra necessário.

Tendo esses dados, é possível refinar a experiência com mais precisão visando as características desejadas. Existem estudos recentes nesse sentido na academia brasileira, o *GLBoard* [38] por exemplo, é um sistema flexível que permite a coleta e análise de dados em jogos educacionais.

Durante a aplicação da pesquisa no semestre 2023.1, como forma de incentivo, o docente ofereceu até 1,0 ponto extra na nota final proporcional ao desempenho dos alunos

nos minigames. Estudos futuros podem explorar o quão isso foi impactante no engajamento dos alunos, comparando a participação nos minigames com alunos que não tiveram o incentivo de nota.

Atualmente, o MagicBits atua como uma ferramenta de **apoio** ao ensino, oferecendo uma forma alternativa para exercitar os conteúdos vistos em sala de aula. Transformar o projeto em uma ferramenta de ensino, onde o aluno adquira os conceitos e a experiência, pode tornar a experiência oferecida pelo MagicBits mais completa e efetiva. Uma vez que o aluno é introduzido a novos conceitos pelo mesmo ambiente e dinâmicas que os exercitam.

Outro ponto de melhoria seria diversificar as artes utilizadas no MagicBits. Pela restrição de tempo e força de trabalho, o MagicBits dependeu da utilização de *assets* de terceiros para o desenvolvimento da parte gráfica e sonora dos minigames. Seria interessante trazer novos contribuidores para o MagicBits e torná-lo um projeto interdisciplinar, com artistas gráficos, músicos e escritores. Dessa forma, seria possível aumentar a fidelidade audiovisual e melhorar a experiência do usuário do jogo, adicionar mais variedade de *assets* e desenvolver narrativas mais elaboradas. Além disso, o projeto ganharia visibilidade, aumentando as chances de surgirem novas aplicações para os sistemas desenvolvidos.

A seguir, uma listagem não-exaustiva de outros pontos de melhoria identificados:

- Pelo *feedback* obtido nos formulários e o que foi discutido na Seção 4.4, é interessante a criação de minigames para o módulo 3 da disciplina;
- Criar *Modo Treino* onde os alunos possam praticar exaustivamente os desafios. Os níveis dos jogos aplicam os desafios, mas de forma parcial e atrelado a narrativa do jogo, geralmente em 3 estágios. Com o *Modo Treino* o aluno poderia escolher um desafio e praticá-lo inúmeras vezes, podendo ter customizações de dificuldade. Com isso, o aluno teria mais oportunidades para exercitar o conteúdo;
- Permitir a seleção do gênero do personagem pelo jogador. Dessa forma, aumentaríamos a representatividade feminina, visto que no momento o personagem do jogador é somente do gênero masculino.

Referências

- [1] Bianca Vargas Tolomei. A Gamificação como Estratégia de Engajamento e Motivação na Educação. *EaD em Foco*, 7(2), September 2017. <https://eademfoco.cecierj.edu.br/index.php/Revista/article/view/440>. 1, 6, 13, 26, 61, 63
- [2] Wikipedia. Nativo digital. *Wikipédia, a enciclopédia livre*, June 2023. https://pt.wikipedia.org/w/index.php?title=Nativo_digital&oldid=66172376. 1
- [3] Thaís Tenório, André Silva, e André Tenório. A influência da gamificação na Educação a Distância com base nas percepções de pesquisadores brasileiros. *EDaPECI*, maio de 2016. <https://seer.ufs.br/index.php/edapeci/article/download/4554/pdf/0>. 1, 11, 79
- [4] Sebastian Deterding, Dan Dixon, Rilla Khaled, e Lennart Nacke. From game design elements to gamefulness: Defining "Gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, MindTrek '11, pages 9–15, New York, NY, USA, 2011. Association for Computing Machinery. <https://doi.org/10.1145/2181037.2181040>. 5, 6
- [5] Fedwa Laamarti, Mohamad Eid, e Abdulmotaleb El Saddik. An overview of serious games. *Int. J. Comput. Games Technol.*, 2014, January 2014. <https://doi.org/10.1155/2014/358152>. 5
- [6] Mélanie Suppan, Gaud Catho, Tomás Robalo Nunes, Valérie Sauvan, Monique Perez, Christophe Graf, Didier Pittet, Stephan Harbarth, Mohamed Abbas, e Laurent Suppan. A Serious Game Designed to Promote Safe Behaviors Among Health Care Workers During the COVID-19 Pandemic: Development of "Escape COVID-19". *JMIR serious games*, 8(4):e24986, December 2020. <https://games.jmir.org/2020/4/e24986>. 5, 10
- [7] Moodle. Moodle plugins - gamification. <https://moodle.org/plugins/index.php>, June 2023. 6
- [8] Christo Dichev e Darina Dicheva. Gamifying education: What is known, what is believed and what remains uncertain: A critical review. *International Journal of Educational Technology in Higher Education*, 14(1):9, December 2017. <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-017-0042-5>. 6, 10, 79

- [9] Karine Heloise Felix de Sousa e Lafayette B. Melo. Uma Revisão Sistemática do Uso da Gamificação no Ensino de Programação. In *Anais do Simpósio Brasileiro de Informática na Educação*, pages 440–450. SBC, November 2021. <https://sol.sbc.org.br/index.php/sbie/article/view/18077>. 6
- [10] Rustici. SCORM Explained 101: One Minute SCORM Overview. <https://scorm.com/scorm-explained/one-minute-scorm-overview/>, 2022. 7, 66
- [11] Wikipedia. E-learning. *Wikipédia, a enciclopédia livre*, December 2022. <https://pt.wikipedia.org/w/index.php?title=E-learning&oldid=64936545>. 7
- [12] Rustici. SCORM Explained: In Depth Review of the SCORM eLearning Standard. <https://scorm.com/scorm-explained/>, 2023. 8
- [13] Victor Gonzalez-Barbone e Luis Anido-Rifon. From SCORM to Common Cartridge: A step forward. *Computers & Education*, 54(1):88–102, January 2010. <https://www.sciencedirect.com/science/article/pii/S0360131509001869>. 8
- [14] MoodleDocs. Moodle - SCORM Supported Versions. https://docs.moodle.org/402/en/SCORM_FAQ#Supported_Versions, 2023. 8
- [15] BlackBoard. SCORM Engine. https://help.blackboard.com/Learn/Administrator/SaaS/Tools_Management/SCORM, 2023. 8
- [16] Instructure. How do I import SCORM files as an assignment? <https://community.canvaslms.com/t5/Instructor-Guide/How-do-I-import-SCORM-files-as-an-assignment/ta-p/673>, July 2020. 8
- [17] John K Haas. A History of the Unity Game Engine. *Diss. Worcester Polytechnic Institute*, 483(2014):484, 2014. <https://core.ac.uk/download/pdf/212986458.pdf>. 8
- [18] Wikipedia. Game Engine. *Wikipedia*, June 2023. https://en.wikipedia.org/w/index.php?title=Game_engine&oldid=1158312991. 8
- [19] mrdoob. Three.js. <https://github.com/mrdoob/three.js>, July 2023. 9
- [20] PixiJS — The HTML5 Creation Engine. PixiJS, July 2023. <https://github.com/pixijs/pixijs>. 9
- [21] Wikipedia. C Sharp (programming language). *Wikipedia*, June 2023. [https://en.wikipedia.org/w/index.php?title=C_Sharp_\(programming_language\)&oldid=1158370695](https://en.wikipedia.org/w/index.php?title=C_Sharp_(programming_language)&oldid=1158370695). 9

- [22] Marcela Pessoa, Rafaela Melo, Gabriel Haydar, David Oliveira, Leandro Carvalho, Elaine Oliveira, Tayana Conte, Filipe Pereira, Luiz Rodrigues, e Seiji Isotani. Uma análise dos tipos de jogadores em uma plataforma de gamificação incorporada a um sistema juiz on-line. In *Anais Do XXXII Simpósio Brasileiro de Informática Na Educação*, pages 474–486, Porto Alegre, RS, Brasil, 2021. SBC. <https://sol.sbc.org.br/index.php/sbie/article/view/18080>. 9
- [23] Wikipedia. Competitive programming. *Wikipedia*, June 2023. https://en.wikipedia.org/w/index.php?title=Competitive_programming&oldid=1159052033. 9
- [24] Yure Oliveira, Claudio Toledo, e Leonardo Pereira. Project Éden: Platform for introductory programming concepts. In *Anais Estendidos Do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 566–573, Porto Alegre, RS, Brasil, 2021. SBC. https://sol.sbc.org.br/index.php/sbgames_estendido/article/view/19690. 10
- [25] Jose Grigorio Neto, Paulo Pimentel Marcolino, Pablo Ferreira, e Daniel Barbosa. Maze code: Retórica procedural aplicada ao ensino de lógica de programação. In *Anais Estendidos Do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 519–528, Porto Alegre, RS, Brasil, 2021. SBC. https://sol.sbc.org.br/index.php/sbgames_estendido/article/view/19685. 10
- [26] Wikipedia. Jogo eletrônico de plataforma. *Wikipédia, a enciclopédia livre*, October 2022. 12
- [27] Wikipedia. Jogo eletrônico de rolagem lateral. *Wikipédia, a enciclopédia livre*, August 2022. 12
- [28] Wikipedia. Jogo eletrônico de ação e aventura. *Wikipédia, a enciclopédia livre*, November 2022. 12
- [29] Wikipedia. Video game developer. *Wikipedia*, October 2022. https://en.wikipedia.org/w/index.php?title=Video_game_developer&oldid=1115222384. 12
- [30] Wikipedia. Game design. *Wikipedia*, June 2023. https://en.wikipedia.org/w/index.php?title=Game_design&oldid=1159554521. 12
- [31] Thiago M. Silva, Leonardo Soares, Katyeudo Karlos de Sousa Oliveira, Rafael Barbosa, Higor Santos, e Gabriel Alves. Experiência com gamificação: Uma abordagem aplicada no ambiente virtual de aprendizagem Moodle. In *Anais do Workshop de Informática na Escola*, pages 677–686. SBC, November 2019. <https://sol.sbc.org.br/index.php/wie/article/view/13216>. 26
- [32] Wikipedia. Fixed-Point Arithmetic. *Wikipedia*, May 2023. https://en.wikipedia.org/w/index.php?title=Fixed-point_arithmetic&oldid=1155743622. 26

- [33] MoodleDocs. Moodle - Open-source Learning Platform | Moodle.org. https://moodle.org/?lang=pt_br, 2022. 65
- [34] Wikipedia. Learning Management System. *Wikipedia*, October 2022. https://en.wikipedia.org/w/index.php?title=Learning_management_system&oldid=1115627398. 65
- [35] Pixel Crushers. Dialogue System for Unity: Dialogue System for Unity. https://www.pixelcrushers.com/dialogue_system/manual2x/html/, 2022. 66
- [36] Randall Fitzgerald. TalkerMakerDeluxe. <https://github.com/digiwombat/TalkerMakerDeluxe>, October 2022. 67
- [37] Wikipedia. Pearson correlation coefficient. *Wikipedia*, July 2023. https://en.wikipedia.org/w/index.php?title=Pearson_correlation_coefficient&oldid=1165632232. 76
- [38] Douglas Silva, Fernanda Pires, Rafaela Melo, e Marcela Pessoa. GLBoard: Um sistema para auxiliar na captura e análise de dados em jogos educacionais. In *Anais Estendidos do XXI Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 959–968, Porto Alegre, RS, Brasil, 2022. SBC. https://sol.sbc.org.br/index.php/sbgames_estendido/article/view/23733. 79

Apêndice A

Formulário de Avaliação da disciplina ISC - 2022.1

Avaliação da disciplina Introdução aos Sistemas Computacionais (ISC) - Semestre 2022.1

18

Respostas

09:00

Tempo médio para concluir

Ativo

Status

1. Quantas matérias você pegou esse semestre? (0 ponto)

18

Respostas

Respostas Mais Recentes

"4"

"3"

"4"

2. Você é aluno transferido de outro curso? (0 ponto)

<input type="radio"/>	Sim	0
<input checked="" type="radio"/>	Não	18



3. Caso seja aluno transferido, selecione a opção correta: (0 ponto)

Transferência externa	0
Transferência interna	0

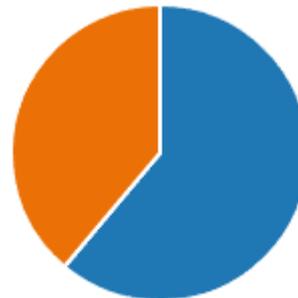
4. Caso seja aluno transferido, qual seria o curso de origem? (0 ponto)

0
Respostas

Respostas Mais Recentes

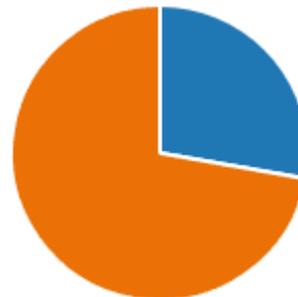
5. Você já tinha conhecimento prévio de programação antes de cursar ISC? (0 ponto)

● Sim	11
● Não	7



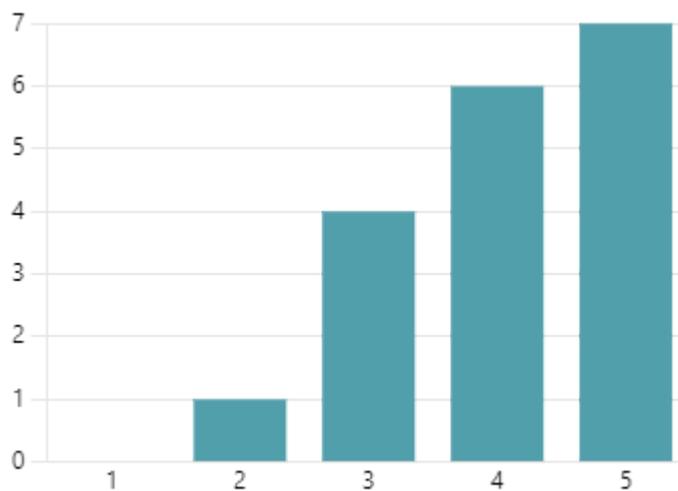
6. Você já tinha conhecimento prévio de números binários antes de cursar ISC? (0 ponto)

● Sim	5
● Não	13



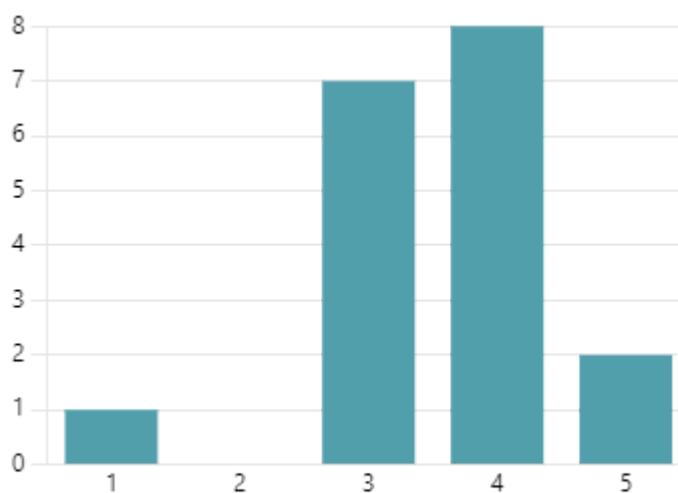
7. O quão difícil você achou ISC? (0 ponto)

4.06
Classificação Média



8. O quão bem você acha que conseguiu absorver o conteúdo de ISC? (0 ponto)

3.56
Classificação Média



9. Qual conteúdo apresentado em ISC você mais gostou? Por quê? (0 ponto)

18
Respostas

Respostas Mais Recentes

"Assembly, pois gosto muito de codar e, até o momento, só h...

"Álgebra de Boole; Módulo 3 (transistores, portas lógicas, circ...

"O módulo sobre assembly, pois embora seja difícil ao meu v...

3 respondentes (17%) responderam **programação** para esta pergunta.



10. Qual conteúdo de ISC você achou mais fácil? Por quê? (0 ponto)

18
Respostas

Respostas Mais Recentes

"Assembly e linguagem de máquina, pois tive mais interesse ...

"Álgebra de Boole e lógica, talvez por ter tido algum contato ...

"O primeiro módulo, porquê de certa forma entendendo o "sc...

6 respondentes (33%) responderam **conteúdo** para esta pergunta.



11. Qual conteúdo de ISC você achou mais difícil? Por quê? (0 ponto)

18
Respostas

Respostas Mais Recentes

"Conversão de bases, pois foi um período de adaptação minh...

"Elétrica, porque eu não havia estudado nada na escola; mon...

"O último módulo, porque ficou bastante abstrato"

5 respondentes (28%) responderam **Assembly** para esta pergunta.



12. Como você acha que a matéria ISC poderia ser melhorada? (Justifique seus pontos.) (0 ponto)

18
Respostas

Respostas Mais Recentes

"A aula e a didática são muito boas, mas acho que falta práti...

"Não me parece ter nenhum problema com a matéria em si, ...

"Acredito que o projeto poderia ter sido apresentado mais ce...

7 respondentes (39%) responderam **conteúdo** para esta pergunta.



13. Tem algum ponto que as questões anteriores não cobriram e você queira acrescentar? (Fique à vontade ;) (0 ponto)

6
Respostas

Respostas Mais Recentes
"Nada :)"

[Atualizar](#)

2 respondentes (29%) responderam **questões** para esta pergunta.

A word cloud visualization of responses. The most prominent words are 'questões' and 'aulas'. Other visible words include 'trabalho', 'valor', 'número', 'jogo', 'professor', 'La', 'Computação', 'ocasiões', 'Ciência da', 'linguagem', 'Assembly', 'menção', 'nota final', 'pequeno deslizo', 'algo', and 'impacto'.

Apêndice B

Formulário de Avaliação da disciplina ISC - 2023.1

Avaliação da disciplina Introdução aos Sistemas Computacionais (ISC) - Semestre 2023.1

8

Responses

06:30

Average time to complete

Active

Status

1. Você é aluno transferido de outro curso? (0 point)

<input type="radio"/>	Sim	0
<input checked="" type="radio"/>	Não	8



2. Caso seja aluno transferido, selecione a opção correta: (0 point)

<input type="radio"/>	Transferência externa	0
<input type="radio"/>	Transferência interna	0

3. Caso seja aluno transferido, qual seria o curso de origem? (0 point)

0

Responses

Latest Responses

4. Você já tinha conhecimento prévio de programação antes de cursar ISC? (0 point)

● Sim 4
● Não 4



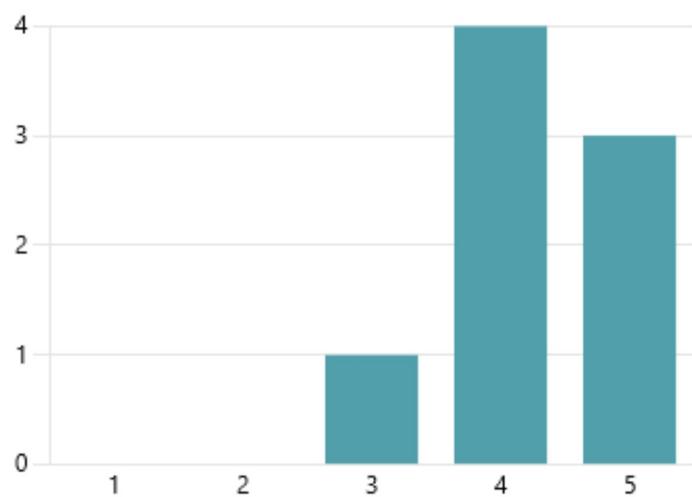
5. Você já tinha conhecimento prévio de números binários antes de cursar ISC? (0 point)

● Sim 4
● Não 4



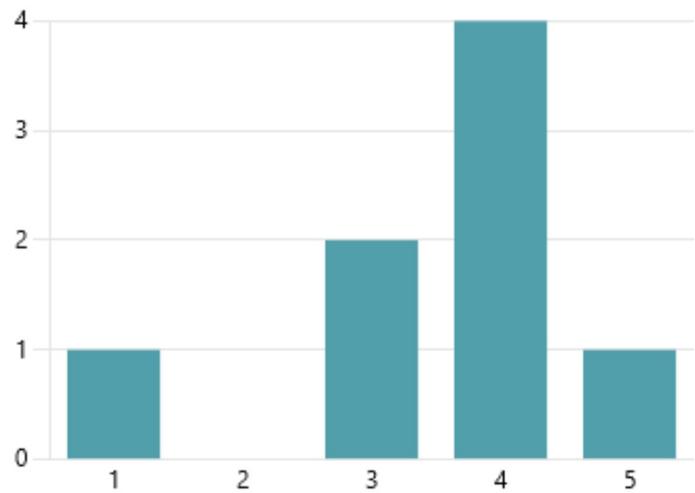
6. O quão difícil você achou ISC? (0 point)

4.25
Average Rating



7. O quão bem você acha que conseguiu absorver o conteúdo de ISC? (0 point)

3.50
Average Rating



8. Qual conteúdo apresentado em ISC você mais gostou? Por quê? (0 point)

8
Responses

Latest Responses

*"O conteúdo de ISC que eu mais gostei foi o do final do Mó...
"assembly, foi o mais pratico que tivemos. Já que tive muita...
"Circuitos lógicos. Gostei da parte elétrica."*

9. Qual conteúdo de ISC você achou mais fácil? Por quê? (0 point)

8
Responses

Latest Responses

*"O conteúdo de ISC que eu achei mais fácil foi o de sistema...
"Nenhum, todos tinham sua dificuldade, seja em acostuma...
"Assembly. A lógica é mais simples do que os outros conteú..."*

10. Qual conteúdo de ISC você achou mais difícil? Por quê? (0 point)

8
Responses

Latest Responses

"O conteúdo de ISC que eu achei mais difícil foi o de eletric...

"Creio que inicialmente foi a conversão de numeros para o...

"O do módulo 1, foi um abalo."

11. Tem algum ponto que as questões anteriores não cobriram e você queira acrescentar? (Fique à vontade ;) (0 point)

3
Responses

Latest Responses

"Foi sem sombras de duvida o curso mais difícil do semestr..."
