



Universidade do Minho
Escola de Engenharia

Tiago Luís Dias da Silva

**Domain-Specific Language
para a criação de Escape Rooms**



Universidade do Minho
Escola de Engenharia

Tiago Luís Dias da Silva

**Domain-Specific Language
para a criação de Escape Rooms**

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho efetuado sob a orientação de
José João Antunes Guimarães Dias de Almeida

Resumo

Esta dissertação de mestrado aborda a criação de uma Domain-Specific Language para a criação de escape rooms. O objetivo é simplificar e aprimorar o processo de concepção e implementação de experiências únicas em escape rooms virtuais. A DSL proposta visa permitir que os criadores desenvolvam cenários, quebra-cabeças e interações de forma rápida e eficiente, sem a necessidade de conhecimento avançado em programação. Além disso, busca promover a reutilização, modularidade, escalabilidade e extensibilidade, oferecendo uma ferramenta especializada para explorar a criatividade de maneira eficiente e flexível. A dissertação inclui uma análise do estado da arte, uma descrição detalhada da metodologia utilizada, resultados preliminares e conclusões relevantes para o desenvolvimento e aplicação de DSLs na criação de jogos de escape.

Palavras-chave Interpretador, Motor de jogos, Linguagem de Domínio Específico, Escape Rooms, Processamento de Linguagens...

Abstract

This master's dissertation addresses the creation of a Domain-Specific Language for the development of escape rooms. The aim is to simplify and enhance the process of designing and implementing unique experiences in virtual escape rooms. The proposed DSL aims to enable creators to develop scenarios, puzzles, and interactions quickly and efficiently, without the need for advanced programming knowledge. Additionally, it seeks to promote reusability, modularity, scalability, and extensibility, providing a specialized tool to explore creativity in an efficient and flexible manner. The dissertation includes an analysis of the state of the art, a detailed description of the methodology used, preliminary results, and relevant conclusions for the development and application of DSLs in escape game creation.

Keywords Interpreter, Engine, Domain-Specific Language, Escape Room, Language Processing...

Conteúdo

1	Introdução	1
1.1	Enquadramento e Motivação	1
1.2	Objetivos	2
1.3	Estrutura da dissertação	3
2	Estado da arte	4
2.1	Escape Rooms	4
2.1.1	Origem, história e evolução	4
2.1.2	Tipologia de escape rooms	6
2.1.3	Ferramentas utilizadas na criação de escape rooms	8
2.2	Domain-Specific Languages	15
2.2.1	Definição e características das DSLs	15
2.2.2	Trabalhos relacionados - DSLs aplicadas na criação de jogos	16
2.2.3	Tecnologia usada no desenvolvimento	17
3	Metodologia	18
3.1	Objetivos	18
3.2	Solução Proposta	19
4	Contribuição	20
4.1	Modelo de uma escape room	20
4.1.1	Mapa	20
4.1.2	Cena	20
4.1.3	Objeto	20
4.1.4	Estado	21
4.1.5	Som	21

4.1.6	Inventário	21
4.1.7	Evento	21
4.2	Gramática da Linguagem	23
4.2.1	Declaração de Variáveis	23
4.2.2	Mapa: Cenas, Objetos e Estados	23
4.2.3	Eventos	24
4.3	Resultados preliminares	27
4.3.1	Exemplo Simples	27
4.3.2	Exemplo Pedir Código	30
4.3.3	Exemplo Arrastar Objeto	31
4.3.4	Editor Gráfico	33
5	Conclusões e trabalho futuro	34
A	Listings	39

Lista de Figuras

1	Pipeline	19
2	Exemplo 1 - Mapa simples (1/5)	28
3	Exemplo 1 - Mapa simples (2/5)	28
4	Exemplo 1 - Mapa simples (3/5)	29
5	Exemplo 1 - Mapa simples (4/5)	29
6	Exemplo 1 - Mapa simples (5/5)	30
7	Exemplo 2 - Pedir código (1/4)	30
8	Exemplo 2 - Pedir código (2/4)	31
9	Exemplo 2 - Pedir código (3/4)	31
10	Exemplo 2 - Pedir código (4/4)	32
11	Exemplo 3 - Arrastar objeto (1/2)	32
12	Exemplo 3 - Arrastar objeto (2/2)	32
13	Exemplo Protótipo Editor Gráfico	33

Acrónimos

2D Duas Dimensões.

3D Três Dimensões.

BGD BoardGameDescription.

DSL Domain-Specific Language.

EBNF Extended Backus-Naur Form.

GPLs General-Purpose Languages.

LALR Look-Ahead Left-to-Right.

REM RoomEscapeMaker.

VR Realidade Virtual.

Capítulo 1

Introdução

1.1 Enquadramento e Motivação

As escape rooms, conforme definidas pelos dicionários ingleses, como Oxford [1] e Cambridge [2], representam jogos cativantes nos quais os participantes encontram-se trancados numa sala temática e têm como objetivo descobrir uma rota de saída, explorando por pistas, desvendando enigmas, solucionando quebra-cabeças e encontrando respostas, tudo isso dentro de um prazo determinado. Apesar de não existir uma tradução oficial nos dicionários portugueses, ocasionalmente são empregados termos como "sala de fuga" ou "sala de escape" para se referir a essas experiências. Contudo, é inegável que o termo em inglês, Escape Room, prevalece.

Na incessante busca por experiências únicas, as escape rooms tradicionais destacam-se como uma atividade bastante popular nos últimos anos nas áreas de lazer, educação e turismo.

Conforme mencionado anteriormente, as escape rooms oferecem experiências singulares, o que as sujeita a desafios locais devido ao potencial esgotamento da audiência local. Nesse contexto, a perspectiva turística surge como uma fonte essencial para revitalização [3, 4].

Na área educacional, as escape room, como elementos de Gamificação (*Gamification*)¹ e podendo ser usadas como aprendizado baseado em jogos (*Game based learning*)², têm sido amplamente testadas e validadas em diversos níveis de escolaridade [5–7] e em uma variedade de disciplinas, incluindo matemática [8, 9], medicina [10, 11], física [12], química [13], biologia [14], programação [15], criptografia [16], inteligência artificial [17] entre outras [18–23]. A aplicação de escape rooms educacionais tem proporcionado experiências imersivas e desafiadoras, estimulando a resolução de problemas, o trabalho em equipe e a aplicação prática do conhecimento em contextos do mundo real. Esse método tem recebido feedback positivo de educadores e estudantes, destacando seu potencial para tornar o processo de

¹ A aplicação de elementos típicos de jogos a outras áreas, como a área educativa.

² Uso de jogos para finalidades didáticas.

aprendizagem mais dinâmico e significativo.

Diante do sucesso das escape rooms tradicionais, a ascensão das tecnologias inevitavelmente deu origem às escape rooms virtuais, expandindo os horizontes desse envolvente entretenimento para uma nova dimensão.

Surge a motivação para explorar uma abordagem inovadora na criação de escape rooms, o desenvolvimento de uma **Domain-Specific Language (DSL)** específica para essa finalidade. A criação de uma DSL dedicada às escape rooms oferece uma oportunidade única de unir o meu interesse pessoal pelo processamento de linguagens e a minha paixão pelas escape rooms virtuais. Com esta linguagem visa-se simplificar e aprimorar o processo de concepção e implementação de novas experiências únicas, proporcionando uma ferramenta especializada para criadores e entusiastas explorarem sua criatividade de maneira eficiente e flexível.

1.2 Objetivos

A tese propõe o desenvolvimento de uma DSL para a criação de escape rooms virtuais, com o intuito de simplificar o processo de concepção e implementação dessas experiências. Os objetivos delineados incluem:

- Identificar e compreender os desafios apresentados em diversos tipos de escape rooms, abrangendo diferentes temáticas e complexidades.
- Pesquisar e analisar diversas ferramentas, softwares e plataformas disponíveis para a criação de escape rooms virtuais. Isso pode envolver a avaliação de soluções existentes, tecnologias emergentes e boas práticas do setor.
- Desenvolver um modelo conceitual que defina as propriedades essenciais de uma escape room, servindo como base conceitual para a linguagem.
- Criar uma gramática que estabeleça as regras e estrutura da linguagem específica para escape rooms. Em seguida, desenvolver um parser capaz de interpretar e processar corretamente os scripts escritos na DSL.
- Desenvolver uma ou mais motores de jogo que, utilizando o output do parser da DSL, sejam capazes de transformar a descrição da escape room em uma aplicação jogável. Isso implica a implementação de mecânicas de jogo, lógica de quebra-cabeças e interatividade.

- Criar uma ferramenta de edição gráfica que facilite a programação na DSL, tornando o processo de criação e modificação de escape rooms mais intuitivo e eficiente. Seria especialmente útil na criação do mapa da escape room e no posicionamento das imagens. Observar que este objetivo é considerado menos prioritário em comparação com os demais, podendo ser deixado de lado caso comprometa o desenvolvimento dos outros.

1.3 Estrutura da dissertação

A estrutura desta dissertação passa pelos seguintes capítulos:

- Capítulo 1: Presente capítulo onde é feita uma contextualização sobre o tema, as motivações presentes na criação de escape rooms e a ideia de como uma DSL seria útil para servir esse propósito. Por fim é feita uma listagem dos objetivos que vão acompanhar esta dissertação.
- Capítulo 2: Neste capítulo será feito um estudo sobre o estado de arte no contexto atual das escape rooms e o desenvolvimento de DSL de maneira ampla.
- Capítulo 3: Neste capítulo descreveremos como pretendemos abordar o desenvolvimento da nossa DSL para a criação de escape rooms
- Capítulo 4: Neste capítulo será relatado o trabalho já realizado no intuito da componente prática do projeto.
- Capítulo 5: Neste capítulo será feita as conclusões e será apresentado o trabalho futuro.

Capítulo 2

Estado da arte

No presente capítulo de estado da arte, almejo situar a pesquisa no contexto atual das escape rooms virtuais, explorando subsequentemente sua concepção e o desenvolvimento de DSL de maneira mais ampla.

2.1 Escape Rooms

Na secção anterior, abordamos uma definição atual elucidativa das escape rooms. No entanto, a natureza altamente versátil desse conceito torna desafiador estabelecer um domínio concreto em torno das escape rooms, dada a variedade de formas e cenários em que essas experiências se manifestam. Diante dessa complexidade, considero crucial explorar a origem, a evolução e a história das escape rooms. Além disso, considero imperativo examinar as diversas tipologias que essas experiências podem assumir, a fim de chegar a um alvo de estudo para o domínio da nossa linguagem. Finalmente, é crucial examinar as soluções contemporâneas já implementadas na criação de escape rooms. Isso permitirá identificar aspectos bem-sucedidos que podem ser incorporados neste projeto, bem como aspetos que podem ser melhorados.

2.1.1 Origem, história e evolução

A história das escape rooms virtuais tem suas raízes no conceito inicial das escape rooms tradicionais, mas sua evolução é única e influenciada por várias fontes. A seguir, seguindo de base o artigo escrito por Nicholson [24], são destacados alguns pontos importantes relacionados à origem e evolução das escape rooms, com foco principal nas virtuais:

- **Real Escape Game em Kyoto (2007).** A primeira atividade documentada chamada de "escape game" foi conduzida pela empresa SCRAP, conhecida como Real Escape Game, em Kyoto,

Japão, em julho de 2007 [25]. O Real Escape Game inicial era uma sala única para equipes de 5-6 jogadores, marcando o início das escape rooms modernas.

- **Expansão Global (2012-2013).** O fenômeno das escape rooms cresceu rapidamente a partir de 2012-2013, começando na Ásia, expandindo-se para a Europa (com destaque para Hungria) e posteriormente alcançando Austrália, Canadá e Estados Unidos.
- **Inspiração nas Mídias Interativas.** Cerca de 65% dos criadores de escape rooms citaram a inspiração em outras escape rooms, como SCRAP¹ do Japão, Parapark² em Budapeste, Hinthunt³ (atualmente fechada) em Londres e Escape the Room⁴ em Nova York. Alguns criadores inspiraram-se em filmes de aventura, como a série de filmes Indiana Jones⁵, e em filmes de terror, como Cube⁶ e Saw⁷.
- **Precursor 1: Live-Action Role-Playing.** Live-Action Role-Playing (LARP), influenciaram o conceito de escape rooms, por fazerem os jogadores quererem experimentar cenários mais imersivos para os seus personagens. Organizações nacionais, como NERO⁸ e IFGS⁹, forneceram cenários imersivos com quebra-cabeças para os participantes resolverem.
- **Precursor 2: Jogos de Aventura Point-and-Click e Escape-The-Room Digital Games.** Estes jogos influenciaram a transição das escape rooms físicas e digitais. Exemplos incluem jogos como Myst¹⁰, que levaram a uma adaptação para jogos digitais onde jogadores tinham de resolver quebra-cabeças para escapar.
- **Precursor 3: Caça ao Tesouro e Puzzle Hunts.** O gênero de caça ao tesouro e puzzle hunts, como o MIT Mystery Hunt¹¹, contribuiu para o desenvolvimento de puzzles mais complexos. Escape rooms incorporaram a ideia de solucionar enigmas em equipa, similar à dinâmica dos puzzle hunts.

¹ <https://realescapegame.jp/about/>

² <https://paraparkbudapest.hu/parapoly>

³ <https://hinthunt.co.uk>

⁴ <https://escapetheroom.com/new-york/rooms/>

⁵ <https://letterboxd.com/film/raiders-of-the-lost-ark/>

⁶ <https://letterboxd.com/film/cube/>

⁷ <https://letterboxd.com/film/saw/>

⁸ <https://www.nerolarp.com>

⁹ <http://www.ifgs.org>

¹⁰ <https://en.wikipedia.org/wiki/Myst>

¹¹ <https://puzzles.mit.edu>

- **Precursor 4: Teatro Interativo e Casas Assombradas.** O crescimento do teatro interativo e das casas assombradas influenciou as escape rooms, especialmente aquelas com temática de horror e interação física com o ambiente.
- **Precursor 5: Shows e Filmes de Aventura.** Séries televisivas, como The Adventure Game¹², e filmes de aventura inspiraram a ideia de resolver enigmas em equipa.
- **Precursor 6: Indústria do Entretenimento Temático.** Escape rooms estão na interseção entre jogos e entretenimento temático, apresentando desafios para negócios comerciais, como restaurantes com mistérios interativos e franquias como 5 Wits¹³.

2.1.2 Tipologia de escape rooms

É necessário explorar a diversidade de experiências oferecidas pelas escape rooms, especialmente quando adaptadas para uma abordagem virtual através de uma DSL. Vamos perceber e discutir como a nossa linguagem pode abordar e incorporar essa diversidade.

Temáticas

As escape rooms podem oferecer diferentes temáticas, como mediaval, futurista, terror, entre outras. Nas escape rooms virtuais, isso é feito através do uso de diferentes imagens e áudios. Por exemplo, criar cenários medievais para uma temática histórica ou introduzir elementos assustadores para uma atmosfera de terror. Tendo isso em conta, acreditamos que a nossa DSL deva permitir importação de assets (áudios e imagens) à escolha do utilizador, de modo a permitir a personalização e expansão das opções temáticas. Pode ainda fornecer templates e kits iniciais especializados em diferentes temáticas.

Formatos

Quanto aos diferentes formatos que as escape rooms podem oferecer, precisamos descartar em primeiro plano as presenciais, pois devido à sua natureza, é impossível entrar no escopo da nossa DSL. Focando apenas nas escape rooms virtuais, vamos deixar de lado, tanto as de **Três Dimensões (3D)** como as de **Realidade Virtual (VR)**, de modo a não elevar demasiado a complexidade do desenvolvimento da linguagem. Sendo assim resta-nos cobrir as escape rooms de **Duas Dimensões (2D)**, que

¹² <https://www.imdb.com/title/tt0283701/>

¹³ <https://5-wits.com>

apesar de parecer pouco, ainda pode vir a se tornar uma ferramenta rica, pela existência de diferentes mecânicas que podem ser abordadas e cobertas pela nossa linguagem.

Mecânicas

Segue em seguida diferentes mecânicas comuns em escape rooms que pensamos cobrir com a nossa DSL.

- **Point-and-Click (Apontar e clicar):** Através de interações simples de apontar e clicar, pode-se fazer os participantes utilizar o rato ou toques na tela para explorar e interagir com elementos do jogo. Acreditamos ser a mecânica mais simples a ser coberta pela nossa linguagem. Exemplo: The Crimson Room¹⁴ é um jogo onde os jogadores exploram um quarto trancado, interagindo com objetos usando cliques para encontrar pistas e resolver enigmas.
- **Drag-and-Drop (Arrastar e Soltar):** Queremos permitir utilizar da mecânica de arrastar e soltar para resolver quebra-cabeças, oferecendo uma abordagem mais tátil e envolvente. Exemplo: The Room¹⁵ é um jogo que possui interações de arrastar e soltar, permitindo que os jogadores manipulem objetos para resolver quebra-cabeças.
- **Interações com Texto:** Permitiremos aos jogadores inserir comandos ou receber informações importantes por meio de diálogos e descrições. Exemplo: Este tipo de interação pode ser encontrado no jogo Zork¹⁶, um dos primeiros jogos de aventura em texto. Nele, os jogadores exploram um mundo virtual através de descrições textuais e interagem com o ambiente digitando comandos de texto, como "ir para o norte" ou "pegar chave".
- **Fluxo Lógico:** Acreditamos ser fulcral permitir a utilização de uma lógica de fluxo, onde a resolução de um enigma leva a outro, construindo uma progressão linear na jogabilidade. Exemplo: Myst¹⁷, é um jogo que incorpora uma progressão lógica, onde a resolução de enigmas específicos abre novas áreas, conduzindo os jogadores pela narrativa.
- **Múltiplos Caminhos:** Para além de um fluxo linear, também queremos permitir oferecer escolhas que impactam o desenrolar da história, proporcionando múltiplos caminhos e desfechos,

¹⁴ <https://www.crazygames.com/game/crimson-room>

¹⁵ https://store.steampowered.com/app/288160/The_Room/

¹⁶ https://store.steampowered.com/app/570580/Zork_Anthology/

¹⁷ <https://en.wikipedia.org/wiki/Myst>

adicionando um elemento de ramificação à narrativa. Exemplo: Her Story¹⁸ é um jogo de mistério onde os jogadores assistem a clipes de vídeo e, com base em suas investigações, moldam a narrativa e descobrem diferentes desfechos.

- **Elementos Cooperativos Online:** A linguagem tem ainda que integrar funcionalidades online para permitir a colaboração entre jogadores, mesmo que estejam fisicamente separados. Exemplo: Keep Talking and Nobody Explodes¹⁹ é um jogo multiplayer em que um jogador visualiza uma bomba enquanto os outros fornecem instruções para desarmá-la.

2.1.3 Ferramentas utilizadas na criação de escape rooms

Existem uma variedade de ferramentas atualmente que abrangem a criação de escape rooms, cada uma oferecendo abordagens distintas para o design e implementação destas experiências envolventes. Ao estudar estas ferramentas vai ser possível prever possíveis características que a nossa DSL terá, bem como algumas abordagens a adotar.

- **Ferramenta de Apresentação Interativa**

Estas ferramentas são comumente utilizadas na Gamificação para fins educacionais, pois, apesar da limitação em mecânicas da experiência criada, oferecem facilidade no desenvolvimento e permitem a criação de experiências imersivas e desafios educativos de forma acessível.

Genially²⁰, Prezi²¹ e Microsoft PowerPoint²² são ferramentas amplamente utilizadas na gamificação para propósitos educacionais, oferecendo facilidade de desenvolvimento e a criação de experiências imersivas e desafios educativos acessíveis. Essas ferramentas destacam-se na construção de escape rooms virtuais, incorporando elementos de gamificação de forma estratégica [26].

O Genially destaca-se pela sua versatilidade na criação de apresentações interativas, proporcionando uma abordagem visualmente rica. Sua aplicação na construção de escape rooms gamificados é notável, permitindo a inclusão de elementos visuais dinâmicos, narrativas não lineares e desafios interativos. Além disso, a plataforma oferece opções tanto gratuitas quanto pagas, focadas especificamente na gamificação e escape rooms. É interessante observar que o Genially já é reconhecido pela comunidade como uma ferramenta eficaz para esse propósito, resultando em

¹⁸ https://store.steampowered.com/app/368370/Her_Story/

¹⁹ <https://keeptalkinggame.com>

²⁰ <https://app.genial.ly>

²¹ <https://prezi.com/pt/>

²² <https://www.microsoft.com/pt-pt/microsoft-365/powerpoint?market=pt>

um maior volume de estudos e modelos dedicados a explorar suas potencialidades nesse contexto [27].

Assim como o Genially, o Prezzi enfatiza uma abordagem narrativa visualmente envolvente. Seu potencial na criação de escape rooms gamificados reside na capacidade de oferecer transições não lineares, elementos visuais interativos e apresentações dinâmicas.

Microsoft PowerPoint surge como uma escolha mais tradicional, proporcionando uma abordagem amplamente conhecida na criação de apresentações interativas. A interface intuitiva e a experiência prévia dos usuários com o PowerPoint facilitam significativamente o processo de criação, tornando-o mais acessível.

- **Scratch.**

O Scratch²³ é uma plataforma de programação visual projetada para facilitar a criação de projetos interativos, jogos e animações. Embora não seja especificamente destinado à criação de Escape Rooms, pode ser uma ferramenta intrigante e educativa para esse propósito. Sua interface amigável e intuitiva torna a plataforma especialmente acessível para iniciantes, incluindo crianças e novatos em programação, que desejam explorar a criação de projetos interativos [28–30].

O ponto forte do Scratch reside na utilização de blocos de programação visual, que podem ser empilhados para criar scripts. Essa abordagem elimina a necessidade de codificação complexa, permitindo que os usuários concentrem-se na lógica em vez da sintaxe da linguagem de programação. Vale ressaltar que o Scratch oferece a flexibilidade de realizar codificação mais avançada e fazer projetos mais ambiciosos, proporcionando uma ponte entre usuários experientes em programação e aqueles que estão dando os primeiros passos. Esta flexibilidade é um ponto positivo que esperamos conseguir obter também na nossa DSL.

A variedade de recursos interativos do Scratch, como detecção de colisão, controle de movimento, efeitos sonoros e animações, pode ser explorada na criação de quebra-cabeças e desafios para um Escape Room. Uma vantagem notável é a comunidade ativa do Scratch, onde os usuários podem compartilhar seus projetos, aprender uns com os outros e até mesmo colaborar em iniciativas relacionadas a Escape Rooms.

- **Cospaces Edu.**

O CoSpaces Edu²⁴ oferece uma abordagem singular na concepção de escape rooms virtuais,

²³ <https://scratch.mit.edu/>

²⁴ <https://www.cospaces.io/>

destacando-se pela criação de ambientes tridimensionais que proporcionam uma imersão única. Embora não seja exclusivamente voltado para escape rooms, sua versatilidade permite a construção de desafiantes e envolventes cenários 3D.

A característica central do CoSpaces Edu reside na capacidade de criar ambientes tridimensionais, explorando diferentes dimensões e perspectivas para aprimorar a experiência do escape room virtual. A ferramenta oferece a inserção de objetos tridimensionais, a criação de animações e a programação de comportamentos interativos.

Similar ao Scratch, o CoSpaces Edu utiliza uma abordagem de programação visual, tornando-o acessível para usuários com diversos níveis de habilidade em programação. Sua funcionalidade de colaboração online facilita a criação conjunta de escape rooms, possibilitando que colaboradores distantes contribuam para o projeto.

Apesar de seus pontos positivos, a curva de aprendizado pode representar um desafio para usuários iniciantes, especialmente aqueles menos familiarizados com a construção de ambientes 3D. Além disso, a versão gratuita do CoSpaces Edu pode apresentar limitações em relação a recursos mais avançados.

A análise dessa ferramenta acabou sendo mais superficial devido à sua ênfase específica em ambientes 3D, uma área que já destacamos anteriormente que não será abordada pela nossa DSL.

- **RoomEscapeMaker.**

O **RoomEscapeMaker (REM)**²⁵, criado por Andre, é descrito pelo próprio criador como um jogo editor grátis de escape rooms online. Por ser um projeto de uma pessoa só e relativamente pequeno, já conta com mais de 30.000 utilizadores e o criador diz que seu projeto já foi usado tanto por professores para fins educacionais, como por empresários para melhorar as experiências dos clientes. Por conta de ser online e da comunidade, é possível jogar as escape rooms criadas por outros, bem como utilizá-las como template ou inspiração.

Decidimos estudar um pouco melhor a parte de edição de escape rooms para entender o que podemos retirar para a nossa linguagem. Primeiro, temos a criação do cenário, que, nesta ferramenta, é feita de forma gráfica, arrastando as imagens para a posição desejada. Apesar de, na nossa DSL, esta parte ter que ser via textual, fica a nota da importância da criação de um editor gráfico de projeção focado principalmente na criação do mapa da escape room.

²⁵ <https://roomescapemaker.com/about>

Falando agora das mecânicas que dão interatividade ao jogo, esta ferramenta baseia-se puramente em eventos 'on click'. Basicamente, a construção dos eventos é: On click action and then show itens and then hide itens. On click while selecting item action and then show itens and then hide itens. On click while selecting any other item ação and then show itens and then hide itens. Sendo os itens todas as imagens colocadas na cena e as actions as seguintes:

- Show small message mensagem
- Reveal secret code código
- Request secret code código
- Add to inventory item
- End game
- Show custom message [Versão paga]
- Show message in popup [Versão paga]
- Request to solve puzzle [Versão paga]

Alguns utilizações exemplo de eventos:

- **Cofre.** Ao colocar as imagens de cofre fechado (visível) e aberto (invisível) e uma chave (invisível) é possível com o evento "On click cofre_fechado Request secret code [code] and then show cofre_aberto, chave and then hide cofre_fechado" criar a interação de um cofre com uma chave lá dentro verdadeiro que só abre com o código certo.
- **Chave no inventário.** Uma noção que este jogo oferece é a de inventário, que basicamente são os itens que o jogador possui no momento. Com isto, é possível fazer o evento "On click Add to inventory [chave] e depois utilizar On click while selecting key para fazer interações com essa chave.
- **Porta.** Não é necessário colocar uma ação para ter uma interação. Por exemplo, se quisermos fazer uma porta abrir com um clique, teríamos que adicionar as imagens da porta fechada (visível) e aberta (invisível) na cena. Em seguida, associar o seguinte evento à porta fechada: On click — and then show porta_aberta and then hide porta_fechada. Se quisermos que seja necessário ter uma chave, basta usar o On click while selecting key."

Ao analisar a funcionalidade do REM, identificamos elementos que podem ser incorporados à nossa DSL. Dentre esses elementos, destacam-se termos como Inventário, Evento, Objeto/Item e

Estado (representando a visibilidade ou invisibilidade da imagem associada ao objeto/item). Além disso, podemos adotar as mecânicas de interação presentes, em que os eventos, principalmente os relacionados a cliques, desencadeiam mudanças de estado nos objetos.

- **Escapp.**

O Escapp²⁶ destaca-se por oferecer um suporte robusto na configuração de escape rooms voltadas para a educação, com a conveniência de logins específicos para professores e alunos. Essa abordagem personalizada reflete o compromisso da ferramenta em atender às necessidades educacionais, proporcionando um ambiente controlado e adaptado para o contexto acadêmico [31].

Se as etapas detalhadas da configuração de uma escape room:

- Básica configuração - Onde se define algumas características da escape room como o título, o assunto, a descrição, a duração, capacidade de equipa e links de suporte.
- Turnos - Onde se define as datas e horas que certos alunos irão poder participar.
- Puzzles - Nesta etapa, especifica-se os diferentes quebra-cabeças que compõem a escape room, cada um constituído por um nome, descrição, resporta, mensagem de sucesso e insucesso e dicas.
- Gerenciamento de dicas - Nesta etapa você pode selecionar a estratégia de dicas da sua sala de fuga. Existem três estratégias disponíveis: não permitir dicas, permitir dicas gratuitas e exigir que os alunos resolvam um questionário para obter uma dica. Se as dicas forem ativas, é possível definir um limite de dicas que os alunos podem obter. Se for escolhida a última estratégia, será necessário fazer upload de um arquivo MoodleXML contendo uma conjunto de perguntas e indicar quantas perguntas serão mostradas em cada tentativa, quantas delas é preciso acertar para passar e se a resposta certa deve ser mostrada para eles após cada tentativa.
- Instruções pré-atividade - Aqui você pode personalizar as informações que os alunos verão antes do início da escape room, desde o momento da inscrição. Por exemplo, é possível lembrá-los de trazer determinado material, pedir que revisem o conteúdo do curso ou fornecer um vídeo para assistir.
- Interface de equipa - Aqui é possível personalizar a interface que os alunos verão em seus computadores quando estiverem a jogar. Pode-se adicionar qualquer informação necessária

²⁶ <https://escapp.es>

para participar: links, imagens, mensagens, a primeira pista, etc.

- Interface da turma - Aqui é possível personalizar a interface que poderá exibir na escape room para todos os alunos verem. Pode incluir um vídeo introdutório, a contagem regressiva, o placar, etc. Se os alunos estiverem a participar remotamente da escape room, eles nunca verão esta interface, então esta etapa pode ser ignorada.
- Avaliação - Nesta etapa personaliza-se um esquema de notas para avaliar a atividade, como a percentagem dada apenas por participar e a percentagem dada a cada quebra-cabeças.

Além disso, a ferramenta oferece um robusto sistema de gerenciamento da escape room, proporcionando uma visão detalhada dos participantes, equipes, resultados, progresso e estatísticas de cada quebra-cabeça. Essas informações são apresentadas de forma intuitiva por meio de gráficos. Com o estudo desta ferramenta, podemos retirar opções de personalização e configuração para a DSL.

- **Engines para a criação de jogos 2D.**

Existem várias engines especializadas na criação de jogos 2D que obviamente podem ser utilizadas para a criação de escape rooms. Ao estudar e analisar essas ferramentas de desenvolvimento de jogos, há implicações valiosas que podem orientar o desenvolvimento da nossa linguagem. Cada ferramenta oferece *insights* únicos que podem ser incorporados no design da DSL, visando aprimorar a experiência do usuário e a eficiência na criação de escape rooms personalizadas.

- Abordagem Sem Programação do GameSalad²⁷:

A abordagem sem programação do GameSalad destaca a importância da usabilidade e acessibilidade. Ao integrar elementos intuitivos na DSL, os utilizador poderão criar escape rooms de forma eficiente, mesmo sem experiência em codificação.

- Agilidade com Blocos de Construção da BuildBox²⁸:

A BuildBox ressalta a agilidade na construção de jogos 2D com a utilização de blocos de construção. Essa ênfase na eficiência pode influenciar o design da DSL, promovendo uma criação rápida e fácil de escape rooms, especialmente para projetos mais simples e casuais, bem como a criação de um editor gráfico especializado em codificação por blocos.

²⁷ <https://gamesalad.com>

²⁸ <https://www.buildbox.com>

- Poder e Flexibilidade com Godot²⁹:

A poderosa engine do Godot e a flexibilidade do GDScript fornecem insights sobre a importância de permitir personalizações avançadas. A DSL pode incorporar opções robustas de codificação para usuários avançados que desejam criar escape rooms altamente personalizadas e complexas.

Ao sintetizar estas influências, a DSL pode ser projetada para atender a uma ampla gama de utilizadores, desde iniciantes até desenvolvedores experientes. A usabilidade intuitiva, agilidade na criação, e o poder de personalização podem ser considerados pilares fundamentais no desenvolvimento da DSL, proporcionando uma ferramenta abrangente na criação de escape room.

- **Implementação da DSL: Pygame (Python) e p5.js (JavaScript)**

Para concretizar as conclusões do estudo sobre diversas ferramentas de criação de escape rooms e incorporar aprendizados na construção da nossa DSL, decidimos por duas implementações distintas com tecnologias bem estabelecidas.

- **Pygame³⁰ (Python³¹):** A primeira versão a ser desenvolvida utilizará o Pygame como base para a implementação em Python. O Pygame é escolhido devido à sua robustez como biblioteca, proporcionando um ambiente propício para a criação de jogos. Vale ressaltar que o parser está sendo desenvolvido em Lark (Python), facilitando a integração entre o script da nossa DSL e a aplicação jogável da escape room.
- **p5.js³² (JavaScript):** Para a versão JavaScript da nossa DSL, decidimos utilizar o p5.js, uma biblioteca que facilita a manipulação de elementos gráficos e interativos em páginas web. A escolha do p5.js visa aproveitar a onipresença do JavaScript na web, proporcionando uma experiência de criação de escape rooms diretamente em navegadores, ampliando a acessibilidade e a distribuição das experiências criadas.

Essas duas versões da DSL serão projetadas com o intuito de proporcionar uma plataforma versátil e abrangente para a criação de escape rooms. Ao alavancar as melhores práticas e características identificadas nas ferramentas estudadas, esperamos oferecer uma DSL que atenda tanto aos iniciantes quanto aos usuários avançados, promovendo a criação eficiente e personalizada de experiências envolventes de escape rooms.

²⁹ <https://godotengine.org>

³⁰ <https://www.pygame.org>

³¹ <https://www.python.org>

³² <https://p5js.org>

2.2 Domain-Specific Languages

Nesta seção, exploraremos as DSLs e o seu papel na criação de soluções especializadas.

Começaremos definindo e destacando as características das DSLs, comparando-as com as **General-Purpose Languages (GPLs)**. Em seguida, examinaremos exemplos de DSLs aplicadas na criação de jogos e em outros domínios semelhantes.

Finalmente, discutiremos a tecnologia que utilizaremos no desenvolvimento da nossa própria DSL para a criação de escape rooms virtuais.

2.2.1 Definição e características das DSLs

As GPLs são projetadas para resolver uma variedade de problemas em diferentes domínios, sendo utilizadas como meio de comunicação entre os programadores e os computadores. Mesmo dentro do campo da programação de propósito geral, existem diversas linguagens, cada uma oferecendo características adaptadas às tarefas específicas em questão, como Python, C ou Java [32]. Diante dessa diversidade, torna-se compreensível a necessidade de linguagens de programação projetadas para resolver problemas específicos em um determinado domínio ou contexto, as chamadas DSLs.

As DSLs diferem das GPLs em várias características que influenciam sua utilidade e implementação. Algumas dessas diferenças incluem:

- **Expressividade:** As DSLs são mais expressivas em relação a um domínio específico, permitindo representar conceitos e operações de forma direta e objetiva. Isso contrasta com as GPLs, que são mais genéricas e abrangentes em sua expressividade.
- **Abstração:** As DSLs oferecem um nível mais alto de abstração, permitindo aos usuários expressarem soluções em termos mais próximos do domínio em questão. Isso contrasta com as GPLs, que podem exigir uma abstração mais baixa e detalhada.
- **Facilidade de compreensão:** As DSLs tendem a ser mais fáceis de compreender, uma vez que são projetadas para refletir o domínio específico, enquanto as GPLs podem ser mais complexas e exigir um conhecimento mais amplo.
- **Especificidade vs. Generalidade:** As DSLs são específicas para um domínio, o que as torna mais limitadas em termos de aplicação, mas mais eficazes dentro desse domínio. Por outro lado, as GPLs são mais gerais e podem lidar com uma variedade mais ampla de problemas, mas podem ser menos eficientes em domínios específicos.

Essas características destacam as diferenças fundamentais entre as DSLs e as GPLs, demonstrando como as DSLs são adaptadas para lidar com problemas específicos de forma mais eficaz e compreensível [32, 33].

As DSLs oferecem várias vantagens em relação às GPLs, incluindo especificidade de domínio, facilidade de uso, facilidade de manutenção e reutilização de código. No entanto, elas também apresentam algumas desvantagens, como limitações de expressividade, custo de desenvolvimento, dificuldade de interoperabilidade e custo de aprendizagem [33].

Em resumo, as DSLs são uma alternativa poderosa às GPLs em muitos casos, proporcionando uma abordagem mais eficaz e intuitiva para resolver problemas específicos em um determinado domínio. No entanto, é importante considerar cuidadosamente as vantagens e desvantagens antes de decidir qual abordagem usar em um determinado projeto.

2.2.2 Trabalhos relacionados - DSLs aplicadas na criação de jogos

Ao investigar exemplos de DSLs aplicadas na criação de jogos, encontramos trabalhos que podem fornecer insights valiosos para o desenvolvimento de uma DSL voltada para a criação de escape rooms. Embora essas DSLs não sejam diretamente destinadas à criação de escape rooms, muitas delas compartilham semelhanças conceituais e técnicas que podem ser adaptadas para atender às necessidades específicas desse contexto.

Na tese "DSL and Engine for Pervasive Treasure Hunt Games"[34] destaca-se a concepção de uma DSL projetada para a criação de jogos de caça ao tesouro. Embora o foco esteja em uma experiência de jogo diferente, podemos retirar a abordagem de simplificar o processo de criação de jogos através de uma DSL, permitindo uma definição intuitiva de locais, desafios e regras do jogo.

Peter Schrotten descreveu uma DSL para expressar jogos de tabuleiro em "Introducing BGD: a DSL to Express Board Games and Gameplay"[35]. A DSL **BoardGameDescription (BGD)** apresenta abstrações específicas para a definição de peças, movimentos e regras em jogos de tabuleiro. Essa abordagem pode ser adaptada para a criação de quebra-cabeças, desafios e mecânicas de jogo em escape rooms, oferecendo uma maneira intuitiva de definir interações e comportamentos dos elementos do jogo.

Em "PULP Scription: A DSL for Mobile HTML5 Game Applications"[36] é projetado uma DSL para a criação de jogos de telemóvel em HTML5. Esta DSL oferece abstrações úteis para a definição de elementos de interface do utilizador e ações como cliques. Estes recursos são relevantes ao criar interfaces interativas como é o desejado neste projeto.

O artigo "A DSL for Rapid Prototyping of Cross-Platform Tower Defense Games"[36] apresenta uma

DSL focada na rápida prototipagem de jogos *Tower Defense*. Mais uma vez é possível retirar princípios de abstração e simplicidade na definição de mapas, inimigos e mecânicas de jogo que podem ser aplicadas na criação de escape rooms.

Os estudos destes exemplos de DSL aplicadas na criação de jogos proporcionaram *insights* valiosos para o desenvolvimento deste projeto. Ao analisar as metodologias, sintaxes e abordagens adotadas nestes exemplos, pudemos identificar elementos-chave que podem ser incorporados neste projeto.

2.2.3 Tecnologia usada no desenvolvimento

Nesta seção, discutiremos a tecnologia utilizada no desenvolvimento da nossa DSL para criação de escape rooms. A principal ferramenta que empregaremos é o Lark, uma biblioteca de análise sintática para Python. O Lark nos permite definir a gramática da nossa linguagem de forma clara e concisa, utilizando a notação **Extended Backus-Naur Form (EBNF)**. Com o Lark, podemos especificar as regras gramaticais da nossa DSL de maneira intuitiva, facilitando o processo de desenvolvimento e garantindo uma sintaxe consistente e compreensível para os usuários.

Optamos por utilizar uma gramática **Look-Ahead Left-to-Right (LALR)** para a nossa DSL devido à sua eficiência na análise de gramáticas complexas e na geração de parsers robustos. A escolha do Lark e da gramática LALR deve-se, em grande parte, ao fato de já termos familiaridade com o uso dessas ferramentas. Essa familiaridade nos permite desenvolver a DSL de forma mais eficiente e rápida, sem descartar a possibilidade de considerar outras ferramentas mais eficientes num estudo futuro.

Capítulo 3

Metodologia

Neste capítulo, descreveremos como pretendemos abordar o desenvolvimento da nossa DSL para a criação de escape rooms. Isso inclui detalhes sobre o processo de desenvolvimento, as etapas a serem seguidas e as ferramentas e tecnologias que serão utilizadas.

3.1 Objetivos

No capítulo anterior, realizou-se um estudo sobre o estado de arte das escape rooms e de trabalhos relacionados com este projeto. Através disso, foi possível delinear objetivos que a nossa linguagem pretende alcançar:

- **Facilitar a criação de escape rooms:** A DSL deve permitir que os criadores de escape rooms desenvolvam cenários, quebra-cabeças e interações de forma rápida e eficiente, sem a necessidade de conhecimento avançado em programação. Isso pode envolver a implementação de abstrações específicas para os elementos comuns encontrados em escape rooms, como cena, objetos, enigmas, etc.
- **Promover a reutilização e modularidade:** A DSL deve facilitar a reutilização de componentes de escape rooms e promover a modularidade do design, permitindo que os criadores compartilhem e combinem diferentes elementos de forma flexível. Isso pode ser alcançado por meio da definição clara de interfaces e abstrações que separam as diferentes partes do jogo.
- **Garantir a escalabilidade e extensibilidade:** A DSL deve ser projetada com a escalabilidade e extensibilidade em mente, permitindo que os criadores de escape rooms adicionem novos recursos e funcionalidades à medida que suas necessidades evoluem. Isso pode ser alcançado por meio de uma arquitetura flexível e modular, que suporte a adição de novos elementos e comportamentos sem alterar o núcleo da linguagem.

3.2 Solução Proposta

A linguagem será desenvolvida em Python, utilizando a biblioteca Lark para a construção do interpretador. A abordagem visa permitir a programação em Python, aproveitando a vasta quantidade de módulos disponíveis para diversos propósitos. A pipeline de execução seguirá um processo simples: o script da nossa linguagem (.er) será analisado pelo parser do Lark e os dados serão armazenados em um dicionário JSON. Diferentes engines poderão então ler este JSON e construir a aplicação da escape room. Atualmente, estamos considerando duas implementações de engines: uma utilizando o módulo Pygame do Python e outra o módulo p5.js do JavaScript.

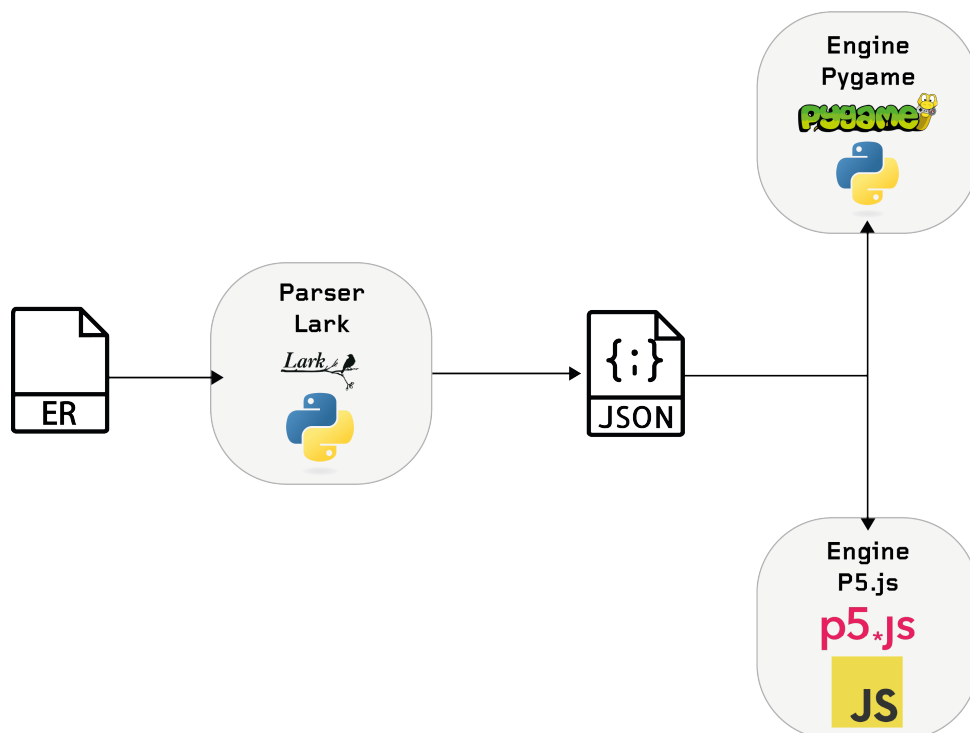


Figura 1: Pipeline

Capítulo 4

Contribuição

Nesta secção, será relatado o trabalho já realizado no intuito da componente prática do projeto. Começando pelo modelo de uma escape room, passando para a gramática da nossa DSL e por fim alguns resultados preliminares.

4.1 Modelo de uma escape room

Nesta seção, apresentaremos o modelo conceitual que define as características essenciais de uma escape room. O modelo é composto por diferentes elementos que descrevem a estrutura e o funcionamento da escape room, incluindo o Mapa, as Cenas, os Objetos, os Estados e os Eventos. Cada um desses componentes desempenha um papel fundamental na criação de uma experiência imersiva para os jogadores.

4.1.1 Mapa

O Mapa representa a estrutura visual da escape room e é composto por uma ou mais cenas. Essas cenas são os ambientes nos quais os jogadores interagem e exploram durante o jogo.

4.1.2 Cena

As Cenas são os ambientes individuais dentro do mapa da escape room. Cada cena é identificada por um ID único, possuir um ou vários estados e pode conter ou não vários objetos.

4.1.3 Objeto

Os Objetos serão os elementos interativos dentro de cada cena. Eles podem representar itens físicos, como chaves e cadeados, ou elementos abstratos, como pistas e quebra-cabeças. Cada objeto é

identificado por um ID único e possui um ou vários estados.

4.1.4 Estado

Os Estados são elementos essenciais que definem as características de um objeto ou cena durante o jogo. Cada Estado é identificado por um ID único e pode incluir informações como imagens associadas, tamanho e posição do objeto ou cena. Eles permitem que os objetos e cenas sejam modificados dinamicamente ao longo do jogo, refletindo diferentes estados e interações. Um Estado pode ter uma ou mais imagens associadas, representando sua aparência visual em um determinado momento. Isso oferece flexibilidade para criar objetos visíveis, invisíveis ou até animações dentro da escape room.

4.1.5 Som

O som não está diretamente associado ao mapa da escape room, mas é necessário para fornecer elementos sonoros que contribuam para a imersão do jogador. Os sons são definidos e identificados de forma semelhante aos outros elementos, utilizando IDs exclusivos para referência.

4.1.6 Inventário

O inventário é um conceito que define os objetos que o jogador possui e que pode usar na cena ou em objetos desta.

4.1.7 Evento

A definição dos Eventos é um pouco mais complexa. Inicialmente, optamos por estruturar os Eventos em pré-condições e pós-condições, como um exemplo típico de uma ação que desencadeia uma mudança de estado em um objeto. No entanto, percebemos que essa abordagem era insuficiente, especialmente quando se tratava de desafios. Portanto, introduzimos a noção de desafios entre as pré-condições e pós-condições. Os desafios são ativados pelas pré-condições e, após sua conclusão, levam a diferentes conjuntos de pós-condições, dependendo do sucesso ou fracasso do desafio.

Para esclarecer a introdução do conceito de desafio, vamos considerar um exemplo de um evento em que a pré-condição seria clicar em um cofre e o desafio associado a essa ação seria solicitar o código do cofre ao jogador. As pós-condições seriam definidas como abrir o cofre em caso de sucesso ou exibir uma mensagem de código incorreto em caso de insucesso. Essa estrutura permite uma abordagem

mais abrangente na definição das interações dentro da escape room, agregando desafios específicos que devem ser superados para avançar no jogo.

As pré-condições podem ser definidas com diversos operadores lógicos, como "ou", "e", "não", entre outros, enquanto as pós-condições são simplesmente listas de ações. As pré-condições funcionam como gatilhos para a ativação dos eventos, enquanto as pós-condições representam as ações a serem executadas após a conclusão de um evento.

A seguir, listamos alguns exemplos de pré-condições, pós-condições e desafios que já foram abordados:

Pré-condições:

- Objeto estar em um estado específico
- Clicar em um objeto
- Não clicar em um objeto
- Clicar em um objeto após a ocorrência de um evento
- Objeto estar em uso

Pós-condições:

- Mudança de estado de um objeto
- Objeto ser adicionado ao inventário
- Finalização do jogo
- Exibição de mensagem
- Mudança de tamanho de um objeto
- Mudança de posição de um objeto
- Transição para outra cena
- Remoção de um objeto
- Reprodução de som

Desafios:

- Pedir um código
- Arrastar um objeto para uma determinada posição

É importante notar que, em trabalho futuro, pretendemos expandir essa lista de desafios, adicionando mais interações e elementos para enriquecer a experiência do jogo.

4.2 Gramática da Linguagem

A gramática da nossa DSL (versão atual em [A.1](#)) para criação de escape rooms permite aos utilizadores declarar variáveis, definir o mapa e os eventos que compõem o jogo.

4.2.1 Declaração de Variáveis

A declaração de variáveis permite atribuir valores a elementos que serão utilizados posteriormente no script. Na nossa DSL, isso é feito da seguinte forma:

```
1 $variavel = valor
```

Neste caso, "variavel" é o nome da variável e "valor" é o valor atribuído a ela.

4.2.2 Mapa: Cenas, Objetos e Estados

Na nossa DSL, a estrutura para definir o mapa é a seguinte:

```
1 Sala id_da_sala [largura, altura] {  
2     contém definição_cena1,  
3     contém definição_cena2,  
4     ...  
5 }
```

A cena possui um estado ou vários e pode conter vários objetos. A estrutura para definir uma cena é a seguinte.

```
1 Cena id_da_cena {  
2     tem definição_estado1,  
3     tem definição_estado2,  
4     ...  
5     tem definição_objeto1,  
6     tem definição_objeto2,  
7     ...  
8 }
```


Os Objetos apesar de ter apenas um ou vários estados, é possível definir a posição e o tamanho e estes serão associados diretamente a todos os estados, ficando com a seguinte estrutura na nossa linguagem:

```
1 Objeto id_do_objeto (x,y) [largura, altura] {
2     tem definição_estado1,
3     tem definição_estado2,
4     ...
5     tem definição_objeto1,
6     tem definição_objeto2,
7     ...
8 }
```

A estrutura base do estado na nossa linguagem é a seguinte:

```
1 Estado id_do_estado "path_da_imagem" (x,y) [largura, altura]
```

No entanto o estado tem algumas peculiaridades, se for adicionado a palavra reservado inicial, este é reconhecido como o estado inicial do objeto ou cena a quem ele foi atribuído.

```
1 Estado inicial id_do_estado "path_da_imagem" (x,y) [largura, altura]
```

Se utilizado a palavra animado, seguido de uma lista de imagens, este é reconhecido como uma animação. É necessário também atribuir mais 2 parâmetros, um que indica de quantos em quantos frames será passado de imagem, e o número de vezes que a animação será repetida.

```
1 Estado animação(frames,repetições) id_do_estado ["path_da_imagem1","path_da_imagem2","
    path_da_imagem3"] (x,y) [largura, altura]
```

Seguem um exemplo de um mapa construído na nossa linguagem:

```
1 Sala sala [1300,700] {
2     contém Cena cena {
3         tem Estado inicial background "room.png"
4         contém Objeto porta (600,342) [225,300] {
5             tem Estado inicial fechada "door.png"
6             tem Estado aberta "open_door.png"
7         }
8         contém Objeto chave (1020,520) [50,25] {
9             tem Estado inicial normal "key.png"
10            tem Estado ativa "active_key.png"
11        }
12    }
13 }
```

4.2.3 Eventos

Os eventos descrevem as interações possíveis na escape room, incluindo ações dos jogadores e suas consequências. Na nossa DSL, a estrutura para definir eventos é a seguinte:

```
1 Evento id_do_evento(repetições):  
2 Se pre-condições,  
3 então pós-condições.
```

As repetições servem para indicar quantas vezes um evento pode ser feito na escape room. Caso não tenha nada, é assumido que esse número é infinito. Caso seja utilizado "-", é assumido como um evento ligado, isto significa, que este é um evento apenas de pós-condições e é chamado em outros eventos que possuem desafios.

```
1 Evento id_do_evento(-):  
2 pós-condições.
```

Falando dos desafios, estes não são representados na estrutura acima que é apenas para os eventos simples. Os eventos que possuem desafios são representados por:

```
1 Evento id_do_evento(repetições):  
2 Se pre-condições,  
3 então desafio.
```

Pré-condições

Objeto estar em um estado específico:

```
1 id_do_objeto está id_do_estado
```

Clicar em um objeto:

```
1 clique id_do_objeto
```

Não clicar em um objeto:

```
1 clique não id_do_objeto
```

Clicar em um objeto após a ocorrência de um evento:

```
1 clique id_do_objeto depois de id_do_evento ter acontecido
```

Objeto estar em uso:

```
1 id_do_objeto está em uso
```

Pós-condições

Mudança de estado de um objeto:

```
1 id_do_objeto muda para id_do_estado
```

Objeto ser adicionado ao inventário:

```
1 id_do_objeto vai para o inventário
```

Finalização do jogo:

```
1 fim de jogo
```

Exibição de mensagem:

```
1 mostra mensagem "texto" em (x,y)
```

Mudança de tamanho de um objeto:

```
1 id_do_objeto muda tamanho para [largura,altura]
```

Mudança de posição de um objeto:

```
1 id_do_objeto muda posição para [x,y]
```

Transição para outra cena:

```
1 muda para cena id_da_cena
```

Remoção de um objeto:

```
1 id_do_objeto é removido
```

Reprodução de som:

```
1 toca id_do_som
```

Desafios

Pedir um código:

```
1 pede código "texto" com mensagem "texto", se acertar faz id_do_evento e se errar faz  
id_do_evento
```

Arrastar um objeto para uma determinada posição:

```
1 arrasta id_do_objeto, se deixar em cima de id_do_objeto faz id_do_evento e se não faz  
id_do_evento
```

Segue em seguida alguns exemplos de eventos definidos na nossa linguagem:

Neste exemplo estão definidos 4 eventos, um que abre a porta quando o jogador tem a chave em uso, outro que acaba com o jogo quando o jogador clica na porta aberta, outro que adiciona a chave ao inventário do jogador quando este clica nela, e por fim um evento que mostra uma mensagem a dizer que a porta está trancada quando o jogador tenta abrir a porta sem chave.

```
1 Evento abrir_porta(1):  
2 Se clique porta e porta está fechada e chave está em uso,  
3 então porta muda para aberta e chave é removida.  
4  
5 Evento fim(1):  
6 se clique porta e porta está aberta,
```

```

7 então fim de jogo.
8
9 Evento chave_inventario(1):
10 Se clique chave,
11 Então chave vai para o inventário.
12
13 Evento mensagem_trancada():
14 Se clique porta e porta está fechada e não chave está em uso, então mostra mensagem "porta
    trancada! procura uma chave!" em (600,385).

```

Neste exemplo pudemos ver o uso do desafio de pedir código. Temos um evento que ao clicar no cofre fechado este pede um código, se o jogador acertar no código, este abre e se não aparece uma mensagem de erro.

```

1 Evento tentar_abrir_cofre():
2 Se clique cofre e cofre está fechado,
3 então pede código "1234" com mensagem "Qual é o código do cofre?" em (1000, 485) , se acertar
    faz abrir_cofre e se errar faz mostrar_erro.
4
5 Evento abrir_cofre(-):
6 cofre muda para aberto e chave muda para normal.
7
8 Evento mostrar_erro(-):
9 mostra mensagem "Código errado!" em (1000, 485).

```

Essa é uma visão geral da gramática da nossa DSL para criação de escape rooms, com base no script fornecido. Nos próximos passos, detalharemos cada construção gramatical e forneceremos exemplos adicionais para ilustrar seu uso na prática.

4.3 Resultados preliminares

Apesar de ainda estar numa fase muito preliminar, a engine em Pygame já está em desenvolvimento e já suporta a criação do mapa bem como a interação de todos os eventos já mostrados em cima. Em seguida mostro alguns exemplos de execução da aplicação gerada em Pygame.

4.3.1 Exemplo Simples

Neste primeiro exemplo veremos a execução de uma simples script escrita na nossa linguagem ([A.2](#)).

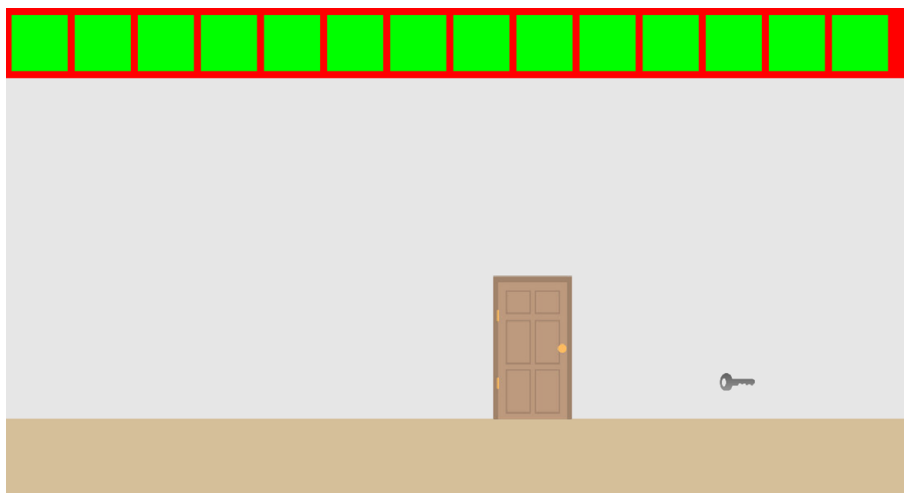


Figura 2: Exemplo 1 - Mapa simples (1/5)

A listra vermelha com quadrados vermelhos na parte de cima da imagem é o inventário do jogador, que no início começa sem objetos. Foi associado um evento que quando o jogador clica na porta e não possui uma chave em uso, esta mostra uma mensagem.

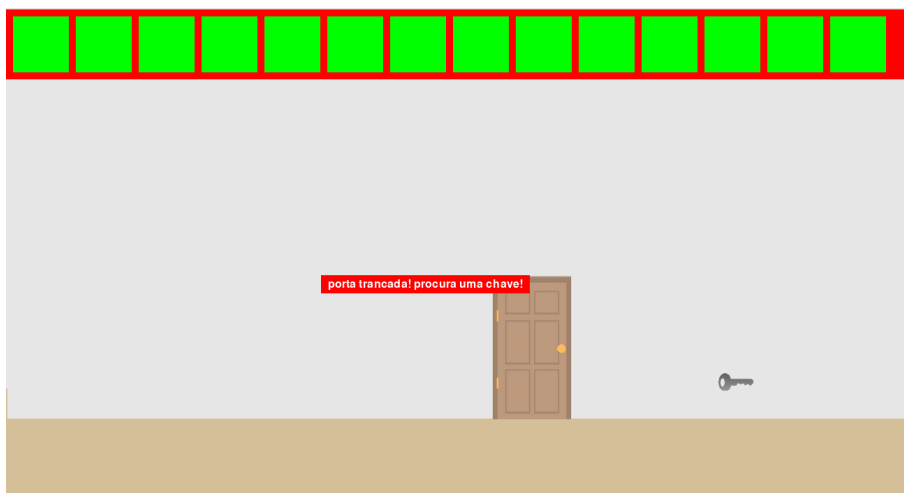


Figura 3: Exemplo 1 - Mapa simples (2/5)

Foi definido um evento que diz que ao clicar na chave esta deve ir para o inventário.

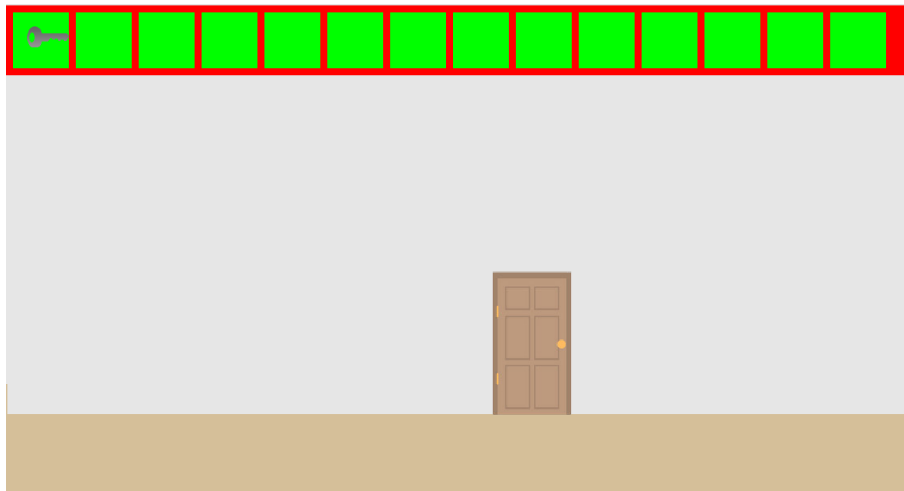


Figura 4: Exemplo 1 - Mapa simples (3/5)

Ao clicar na chave esta fica ativa, e por sua vez, é possível clicar na porta. Agora com a chave em uso, o evento manda abrir a porta.

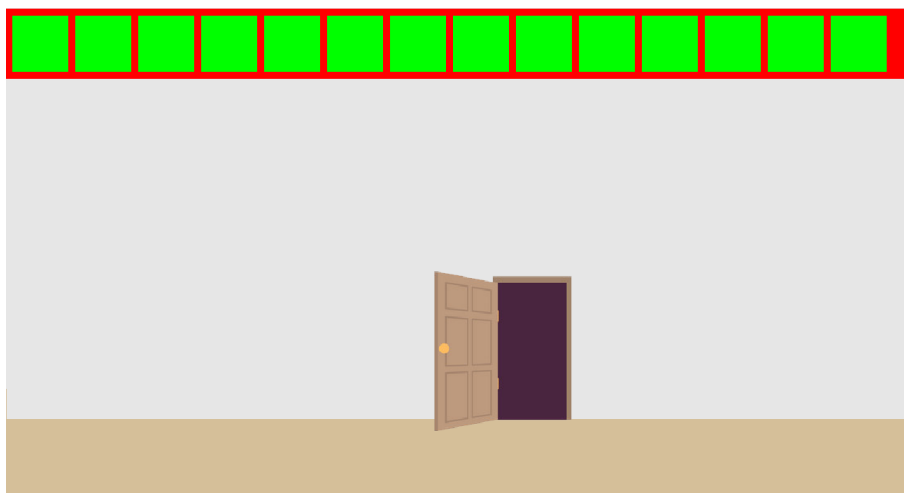


Figura 5: Exemplo 1 - Mapa simples (4/5)

Por fim, ao clicar na porta aberta o jogo termina.

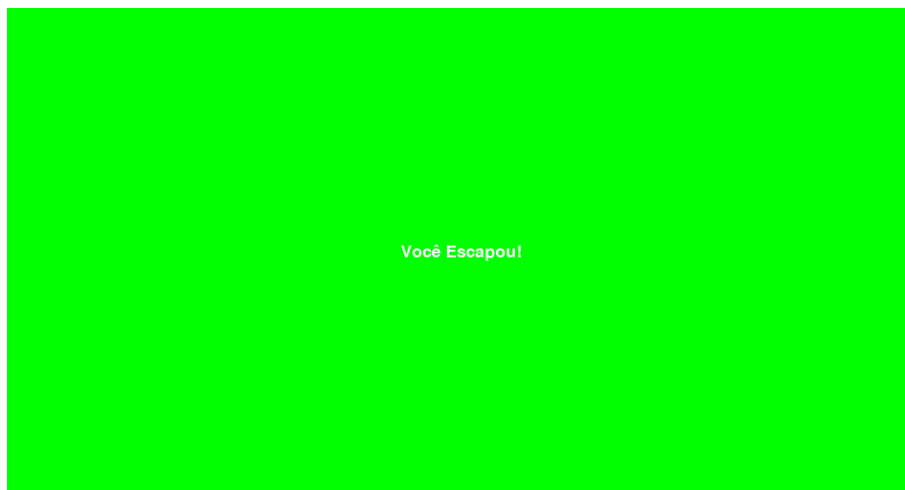


Figura 6: Exemplo 1 - Mapa simples (5/5)

4.3.2 Exemplo Pedir Código

Neste exemplo vamos ver a execução do exemplo do desafio de pedir código (A.3) que está associado a um cofre.

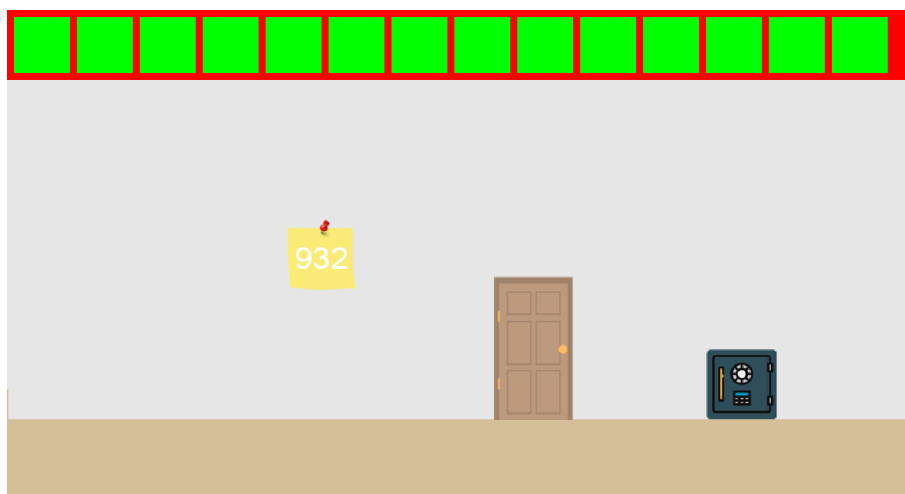


Figura 7: Exemplo 2 - Pedir código (1/4)

Neste exemplo, a cena contém um cofre e ao clicar nele podemos ver que nos pede um código para abrir.

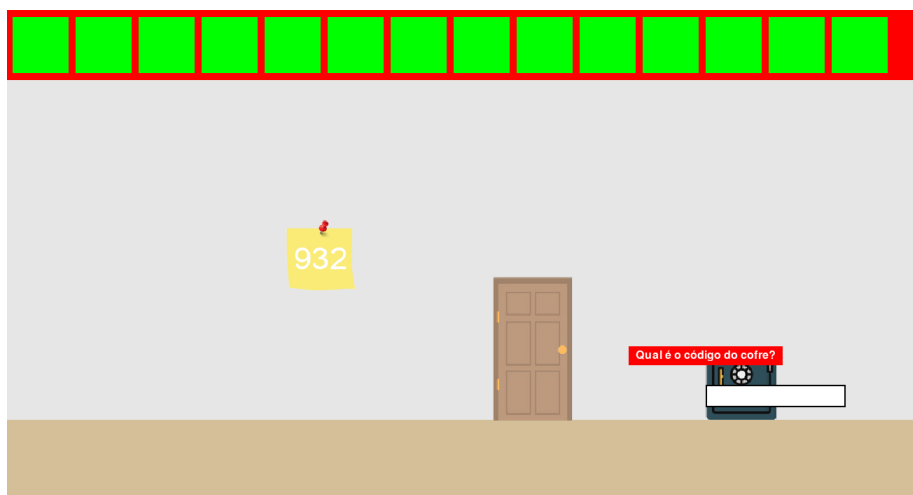


Figura 8: Exemplo 2 - Pedir código (2/4)

Ao colocar o código errado este exibe uma mensagem de erro.

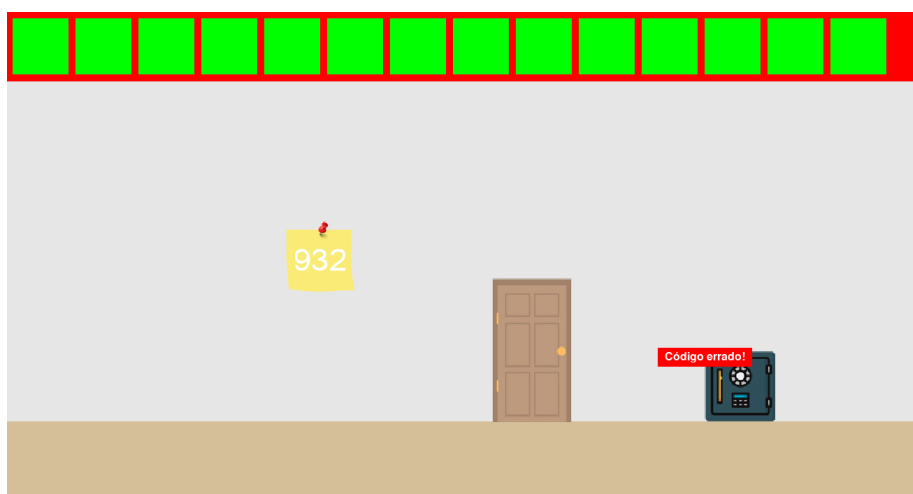


Figura 9: Exemplo 2 - Pedir código (3/4)

Se colocar o código certo, o código que está na nota à esquerda da porta, o cofre abre e revela a chave dentro dele.

4.3.3 Exemplo Arrastar Objeto

Neste exemplo vamos poder ver a execução de um exemplo do desafio arrastar objeto visto no (A.4). Este exemplo mostra uma lupa no mapa que quando é arrastada para cima de uma nota, esta é ampliada podendo ser visível o conteúdo escrito nela.

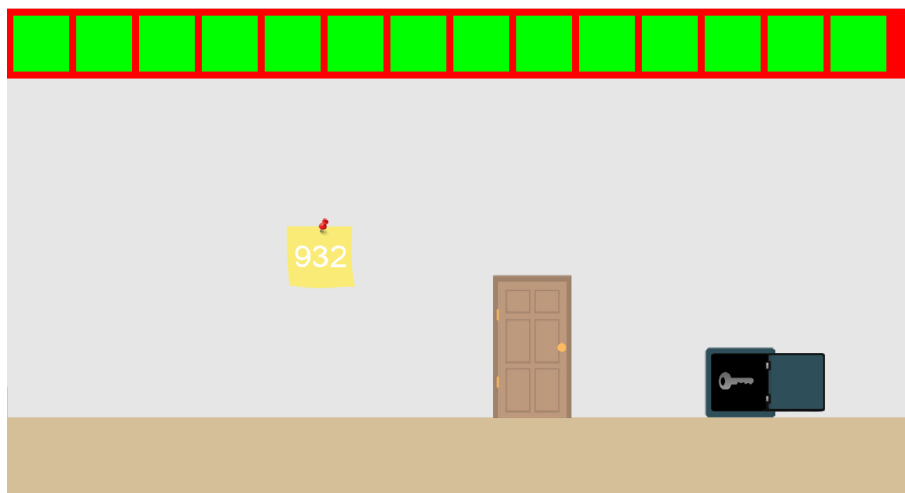


Figura 10: Exemplo 2 - Pedir código (4/4)

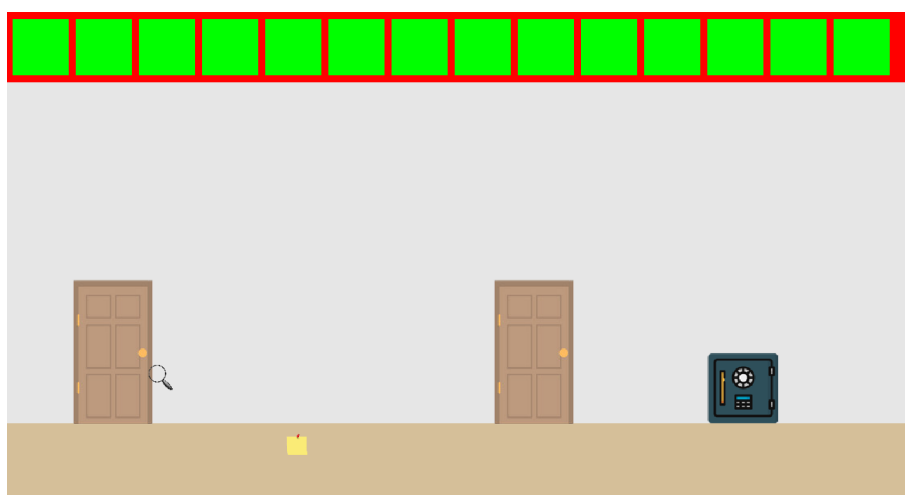


Figura 11: Exemplo 3 - Arrastar objeto (1/2)

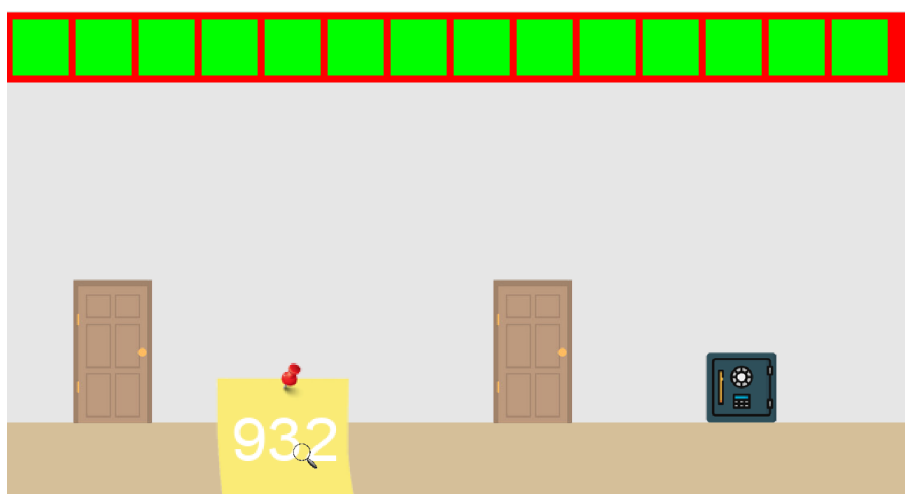


Figura 12: Exemplo 3 - Arrastar objeto (2/2)

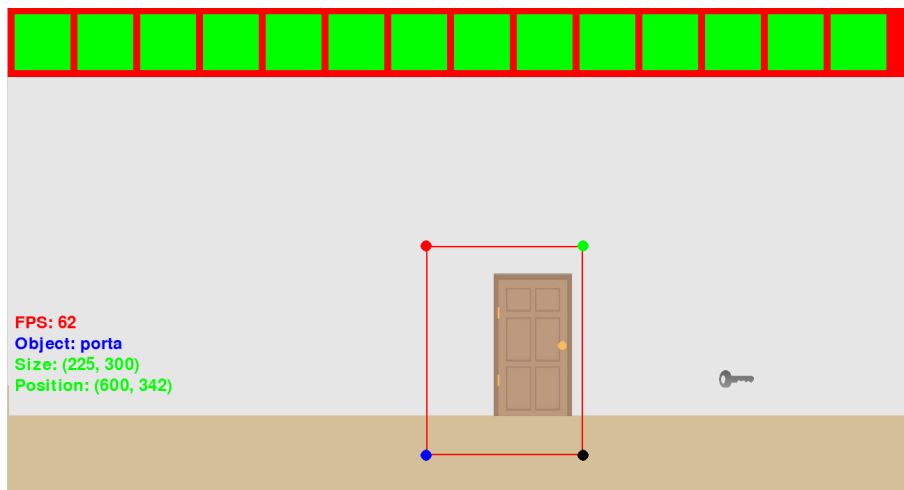


Figura 13: Exemplo Protótipo Editor Gráfico

4.3.4 Editor Gráfico

Já existe também um protótipo do que seria o editor gráfico para ajudar na construção dos mapas da escape rooms, feito ele também em Pygame. Na figura de baixo podemos ver que este mostra os objetos na tela e ao interagir com eles apresenta à esquerda a sua posição e tamanho. É possível arrastar os objetos para mudar a posição, bem como arrastando os pontos das extremidades do perímetro do objeto aumentar e diminuir o tamanho deste.

Capítulo 5

Conclusões e trabalho futuro

Nesta fase do projeto já conseguimos obter alguns resultados preliminares satisfatórios, mostrando que estamos a fazer um bom trabalho que a nossa DSL caminha num bom caminho, no entanto ainda existe algum trabalho futuro a ser feito, como adicionar mais desafios à nossa linguagem, fazer uma engine em p5.js de modo a ser possível jogar na web, melhorar o editor de projeção gráfico de modo a facilitar o desenvolvimento de escape rooms na nossa escape rooms, entre outros trabalhos que serão descobertos futuramente.

Bibliografia

- [1] *escape-room noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com*. URL: <https://www.oxfordlearnersdictionaries.com/definition/english/escape-room> (acedido em 15/01/2024) (ver p. 1).
- [2] *escape room*. pt. Jan. de 2024. URL: <https://dictionary.cambridge.org/pt/dicionario/ingles/escape-room> (acedido em 15/01/2024) (ver p. 1).
- [3] A. V. Lama. «Millennial leisure and tourism: The rise of escape rooms». Em: *Cuadernos de Turismo* 41 (2018), pp. 743–746 (ver p. 1).
- [4] Farkhondeh Fazel Bakhsheshi. «Escape Rooms: A New phenomenon in Tourism». Em: *2019 International Serious Games Symposium (ISGS)*. 2019, pp. 38–41. DOI: [10.1109/ISGS49501.2019.9047040](https://doi.org/10.1109/ISGS49501.2019.9047040) (ver p. 1).
- [5] Lafayette Batista Melo. «Playing with Escape Room Games for Children : An online experience at home in times of pandemic». Em: *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*. 2021, pp. 1–6. DOI: [10.23919/CISTI52073.2021.9476343](https://doi.org/10.23919/CISTI52073.2021.9476343) (ver p. 1).
- [6] Samantha Clarke et al. «EscapED: A Framework for Creating Educational Escape Rooms and Interactive Games to For Higher/Further Education.» Em: *International Journal of Serious Games* 4 (set. de 2017). DOI: [10.17083/ijsg.v4i3.180](https://doi.org/10.17083/ijsg.v4i3.180) (ver p. 1).
- [7] Harsha Kurian e Robert Ross. «Educational Escape Room for an Interactive University Class Room». Tese de doutoramento. Mar. de 2021 (ver p. 1).
- [8] Jake Andrews e Ovidiu Bagdasar. Em: *Open Education Studies* 5.1 (2023), p. 20220194. DOI: [doi:10.1515/edu-2022-0194](https://doi.org/10.1515/edu-2022-0194). URL: <https://doi.org/10.1515/edu-2022-0194> (ver p. 1).
- [9] Cristina Jiménez et al. «Digital Escape Room, Using Genial.Ly and A Breakout to Learn Algebra at Secondary Education Level in Spain». Em: *Education Sciences* 10.10 (2020). ISSN: 2227-7102.

- DOL: [10.3390/educsci10100271](https://doi.org/10.3390/educsci10100271). URL: <https://www.mdpi.com/2227-7102/10/10/271> (ver p. 1).
- [10] Ahmad Neetoo, Nabil Zary e Youness Zidoun. *Escape Rooms in Medical Education: Deductive Analysis of Designs, Applications and Implementations*. Jul. de 2021. DOI: [10.20944/preprints202107.0644.v1](https://doi.org/10.20944/preprints202107.0644.v1) (ver p. 1).
- [11] «Escape MD: Using an Escape Room as a Gamified Educational and Skill-Building Teaching Tool for Internal Medicine Residents». Em: 13 (). ISSN: 2168-8184. DOI: [10.7759/cureus.18314](https://doi.org/10.7759/cureus.18314). URL: <http://dx.doi.org/10.7759/cureus.18314> (ver p. 1).
- [12] Alpár Vörös e Susana Sárközi. «Physics escape room as an educational tool». Em: vol. 1916. Dez. de 2017, p. 050002. DOI: [10.1063/1.5017455](https://doi.org/10.1063/1.5017455) (ver p. 1).
- [13] Aleksandra Naumoska, Hari Dimeski e Marina Stojanovska. «Using the Escape Room game-based approach in chemistry teaching». Em: *Journal of the Serbian Chemical Society* 88 (jan. de 2022), pp. 88–88. DOI: [10.2298/JSC211228088N](https://doi.org/10.2298/JSC211228088N) (ver p. 1).
- [14] Stylianos Mystakidis, Enrique Cachafeiro e Ioannis Hatzilygeroudis. «Enter the Serious E-scape Room: A Cost-Effective Serious Game Model for Deep and Meaningful E-learning». Em: *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*. 2019, pp. 1–6. DOI: [10.1109/IISA.2019.8900673](https://doi.org/10.1109/IISA.2019.8900673) (ver p. 1).
- [15] Sonsoles López-Pernas et al. «Examining the Use of an Educational Escape Room for Teaching Programming in a Higher Education Setting». Em: *IEEE Access* 7 (2019), pp. 31723–31737. DOI: [10.1109/ACCESS.2019.2902976](https://doi.org/10.1109/ACCESS.2019.2902976) (ver p. 1).
- [16] Stefan Seebauer, Sabrina Jahn e Jürgen Mottok. «Learning from Escape Rooms? A Study Design Concept Measuring the Effect of a Cryptography Educational Escape Room». Em: *2020 IEEE Global Engineering Education Conference (EDUCON)*. 2020, pp. 1684–1685. DOI: [10.1109/EDUCON45650.2020.9125333](https://doi.org/10.1109/EDUCON45650.2020.9125333) (ver p. 1).
- [17] Daniella DiPaola et al. «DRONEscape: Designing an Educational Escape Room for Adult AI Literacy». Em: *2023 IEEE Conference on Games (CoG)*. 2023, pp. 1–8. DOI: [10.1109/CoG57401.2023.10333148](https://doi.org/10.1109/CoG57401.2023.10333148) (ver p. 1).
- [18] Sandra Calvo et al. «ROOM ESCAPE: A TRANSVERSAL GAMIFICATION STRATEGY FOR PHYSIOTHERAPY STUDENTS». Em: jul. de 2018. DOI: [10.21125/edulearn.2018.1051](https://doi.org/10.21125/edulearn.2018.1051) (ver p. 1).

- [19] Aisyah Saad Abdul Rahim et al. «Educational escape rooms in pharmacy education: A narrative review». Em: *Pharmacy Education* 22.1 (jun. de 2022), p. 540–557. DOI: [10.46542/pe.2022.221.540557](https://doi.org/10.46542/pe.2022.221.540557). URL: <https://pharmacyeducation.fip.org/pharmacyeducation/article/view/1659> (ver p. 1).
- [20] Briyana Morrell e Heidi N Eukel. «Shocking Escape: A Cardiac Escape Room for Undergraduate Nursing Students». Em: *Simulation & Gaming* 52.1 (fev. de 2021). Publisher: SAGE Publications Inc, pp. 72–78. ISSN: 1046-8781. DOI: [10.1177/1046878120958734](https://doi.org/10.1177/1046878120958734). URL: <https://doi.org/10.1177/1046878120958734> (acedido em 15/01/2024) (ver p. 1).
- [21] Aldo Gordillo et al. «Evaluating an Educational Escape Room Conducted Remotely for Teaching Software Engineering». Em: *IEEE Access* 8 (dez. de 2020), pp. 225032–225051. DOI: [10.1109/ACCESS.2020.3044380](https://doi.org/10.1109/ACCESS.2020.3044380) (ver p. 1).
- [22] Ladislav Huraj, Roman Hrmo e Marianna Sejutová Hudáková. «The Impact of a Digital Escape Room Focused On HTML and Computer Networks on Vocational High School Students». Em: *Education Sciences* 12 (out. de 2022), p. 682. DOI: [10.3390/educsci12100682](https://doi.org/10.3390/educsci12100682) (ver p. 1).
- [23] Mário Cruz. «'Escapando de la clase tradicional': the escape rooms methodology within the spanish as foreign language classroom». Em: *Revista Lusófona de Educação* 46 (dez. de 2019), pp. 117–137. DOI: [10.24140/issn.1645-7250.rle46.08](https://doi.org/10.24140/issn.1645-7250.rle46.08) (ver p. 1).
- [24] Scott Nicholson. *Peeking behind the locked door: A survey of escape room facilities*. White Paper. Available at <http://scottnicholson.com/pubs/erfacwhite.pdf>. 2015 (ver p. 4).
- [25] *What is Real Escape Game? | Real Escape Game in Japan | Created by SCRAP*. URL: <https://realescapegame.jp/about/> (acedido em 15/01/2024) (ver p. 5).
- [26] Mayra I. Barrera G, Daniela Benalcázar Ch. e Dra. Carolina San Lucas S. «Gamification in the teaching of prevention measures for Covid-19». Em: *2022 IEEE Global Engineering Education Conference (EDUCON)*. 2022, pp. 1512–1516. DOI: [10.1109/EDUCON52537.2022.9766821](https://doi.org/10.1109/EDUCON52537.2022.9766821) (ver p. 8).
- [27] Ximena Hernández et al. «Genially gamification tool for teaching and learning Mathematics». Em: *Revista Imaginario Social* 7 (jan. de 2024). DOI: [10.59155/is.v7i1.151](https://doi.org/10.59155/is.v7i1.151) (ver p. 9).
- [28] Melissa Ford. 2017 (ver p. 9).

- [29] Kashif Amanullah e Tim Bell. «Analysing Students' Scratch Programs and Addressing Issues using Elementary Patterns». Em: *2018 IEEE Frontiers in Education Conference (FIE)*. 2018, pp. 1–5. DOI: [10.1109/FIE.2018.8658821](https://doi.org/10.1109/FIE.2018.8658821) (ver p. 9).
- [30] Chiş Lavinia-Maria. «Case Study of Learning Music using Scratch Programming Environment and the Impact of Musical Knowledge on Learning Computer Programming». Em: *2021 16th International Conference on Engineering of Modern Electric Systems (EMES)*. 2021, pp. 1–4. DOI: [10.1109/EMES52337.2021.9484148](https://doi.org/10.1109/EMES52337.2021.9484148) (ver p. 9).
- [31] Sonsoles López-Pernas et al. «Escapp: A Web Platform for Conducting Educational Escape Rooms». Em: *IEEE Access* 9 (2021), pp. 38062–38077. DOI: [10.1109/ACCESS.2021.3063711](https://doi.org/10.1109/ACCESS.2021.3063711) (ver p. 12).
- [32] Markus Voelter. «From General Purpose Languages to DSLs». Em: *DSL Engineering: Designing, Implementing and Using Domain-Specific Languages*. Ed. por Markus Voelter. CreateSpace Independent Publishing Platform, 2013. Cap. 2.2, pp. 25–31 (ver pp. 15, 16).
- [33] Nuno Oliveira et al. «Domain specific languages: a theoretical survey». Em: *INFORUM'09 Simpósio de Informática*. Faculdade de Ciências da Universidade de Lisboa. 2009. URL: <http://hdl.handle.net/10198/1192> (ver p. 16).
- [34] Habibollah Hosseinpour e Christian Skar. «DSL and Engine for Pervasive Treasure Hunt Games». Em: 2012. URL: <https://api.semanticscholar.org/CorpusID:108176915> (ver p. 16).
- [35] Peter Schroten. *Introducing BGD: a DSL to express board games and gameplay*. Mai. de 2019. URL: <http://essay.utwente.nl/79157/> (ver p. 16).
- [36] Kevin Sánchez, Kelly Garcés e Rubby Casallas. «A DSL for rapid prototyping of cross-platform tower defense games». Em: *2015 10th Computing Colombian Conference (10CCC)*. 2015, pp. 93–99. DOI: [10.1109/ColumbianCC.2015.7333417](https://doi.org/10.1109/ColumbianCC.2015.7333417) (ver p. 16).


```

33         | ID "está" ID                                -> esta_estado
34         | "clique" ID "depois de" ID "ter acontecido" -> clique_depois
35         | ID "está em uso"                             -> uso
36
37 poscondicoes : poscondicao ("e" poscondicao)*
38 poscondicao   : ID "muda para" ID
39               -> muda_estado
40               | ID "vai para o inventário"
41               -> vai_inv
42               | "fim de jogo"
43               -> fim_de_jogo
44               | "mostra mensagem" text "em" posicao
45               -> mensagem
46               | "pede código" text "com mensagem" text "em" posicao ", se acertar faz" ID "e
47 se errar faz" ID -> pede_codigo
48               | "arrasta" ID ", se deixar em cima de" ID "faz" ID "e se não faz" ID
49               -> arrasta_objeto
50               | ID "muda tamanho para" tamanho
51               -> muda_tamanho
52               | ID "muda posição para" posicao
53               -> muda_posicao
54               | "muda para cena" ID
55               -> muda_cena
56               | ID "é removido" ("o"|"a")
57               -> remove
58               | "toca" ID
59               -> toca_som
60
61 posicao        : posicao_value
62               | ARG                                -> posicao_arg
63 posicao_value  : "(" par ")"
64 tamanho       : tamanho_value
65               | ARG                                -> tamanho_arg
66 tamanho_value : "[" par "]"
67 par           : numero "," numero
68 filenames     : filename ("," filename)*
69 filename       : text
70 text          : TEXTO
71               | ARGN -> text_argn
72               | ARG  -> text_arg
73
74 numero        : NUM
75               | ARG -> num_arg
76
77 color         : "(" numero "," numero "," numero ")"
78
79 value         : TEXTO                                -> value_text
80               | posicao_value                        -> value_posicao
81               | tamanho_value                       -> value_tamanho

```

```

71         | NUM                                -> value_num
72
73
74 INICIAL : /inicial/
75 REVERSE : /\.reverse\(\)/
76
77 ID      : /\w\[_]+\//
78 TEXTO   : /"[^"]*"//
79 NUM     : /\d+/
80 ARGN    : /\$\d+/
81 ARG     : /\$[a-zA-Z]\w+/
82
83 %import common.WS
84 %ignore WS

```

Listing A.1: Gramática atual da nossa DSL

```

1 Sala sala [1300,700] {
2     contém Cena cena {
3         tem Estado inicial background "room.png"
4         contém Objeto porta (600,342) [225,300] {
5             tem Estado inicial fechada "door.png"
6             tem Estado aberta "open_door.png"
7         }
8         contém Objeto chave (1020,520) [50,25] {
9             tem Estado inicial normal "key.png"
10            tem Estado ativa "active_key.png"
11        }
12    }
13 }
14
15 #Eventos:
16 Evento abrir_porta(1):
17 Se clique porta e porta está fechada e chave está em uso,
18 então porta muda para aberta e chave é removida.
19
20 Evento fim(1):
21 se clique porta e porta está aberta,
22 então fim de jogo.
23
24 Evento chave_inventario(1):
25 Se clique chave,
26 Então chave vai para o inventário.
27
28 Evento mensagem_trancada():
29 Se clique porta e porta está fechada e não chave está em uso,
30 então mostra mensagem "porta trancada! procura uma chave!" em (600,385).

```

Listing A.2: Exemplo 1 de uma script na nossa DSL

```

1 $room = "room.png"
2 $tamanho = [225,300]
3 $posicao = (1000,485)
4 $num = 100
5 $codigo = "932"
6
7 Sala sala [1300,700] {
8     contém Cena cena {
9         tem Estado inicial background $room
10        contém Objeto porta (600,342) $tamanho {
11            tem Estado inicial fechada "door.png"
12            tem Estado animado(3,1) aberta ["door/door1.png","door/door2.png","door/door3.png",
13            "door/door4.png","door/door5.png"]
14        }
15        contém Objeto cofre $posicao [172,$num] {
16            tem Estado inicial fechado "cofre.png"
17            tem Estado aberto "open_cofre.png"
18        }
19        contém Objeto chave (1020,520) [50,25] {
20            tem Estado normal "key.png"
21            tem Estado ativa "active_key.png"
22        }
23        contém Objeto nota {
24            tem Estado inicial normal "nota_nova.png" (400,300) [100,100]
25        }
26    }
27
28 #Eventos:
29
30 Evento abrir_porta(1):
31 Se clique porta e porta está fechada e chave está em uso,
32 então porta muda para aberta e chave é removida.
33
34 Evento fim(1):
35 se clique porta e porta está aberta,
36 então fim de jogo.
37
38 Evento chave_inventario(1):
39 Se clique chave e cofre está aberto,
40 Então chave vai para o inventário.
41
42 Evento tentar_abrir_cofre():
43 Se clique cofre e cofre está fechado,
44 então pede código $codigo com mensagem "Qual é o código do cofre?" em (1000, 485) , se acertar
45     faz abrir_cofre e se errar faz mostrar_erro.
46
47 Evento abrir_cofre(-):
48 cofre muda para aberto e chave muda para normal.

```

```

48
49 Evento mostrar_erro(-):
50 mostra mensagem "Código errado!" em (1000, 485).
51
52 Evento mensagem_trancada():
53 Se clique porta e porta está fechada e não chave está em uso,
54 então mostra mensagem "porta trancada! procura uma chave!" em (600,385).

```

Listing A.3: Exemplo 2 de uma script na nossa DSL

```

1 $room = "room.png"
2 $tamanho = [225,300]
3 $posicao = (1000,485)
4 $num = 100
5 $codigo = "932"
6
7 Sala sala [1300,700] {
8     contém Cena cena {
9         tem Estado inicial background $room
10        contém Objeto porta (600,342) $tamanho {
11            tem Estado inicial fechada "door.png"
12            tem Estado animado(3,1) aberta ["door/door1.png","door/door2.png","door/door3.png",
13            "door/door4.png","door/door5.png"]
14        }
15        contém Objeto porta2 (0,342) $tamanho {
16            tem Estado inicial fechada "door/door0.png"
17            tem Estado animado(3,1) aberta glob("doort/*.png")
18        }
19        contém Objeto cofre $posicao [172,$num] {
20            tem Estado inicial fechado "cofre.png"
21            tem Estado aberto "open_cofre.png"
22        }
23        contém Objeto chave (1020,520) [50,25] {
24            tem Estado normal "key.png"
25            tem Estado ativa "active_key.png"
26        }
27        contém Objeto nota {
28            tem Estado inicial normal "nota.png" (400,600) [30,30]
29        }
30        contém Objeto nota_lupa (300,500) [200,200] {
31            tem Estado visivel image_plus_txt("nota.png","nota_nova.png",$codigo,"arial.ttf",
32            (255,255,255))
33        }
34        contém Objeto lupa (200,500) [40,40] {
35            tem Estado inicial normal "lupa.png"
36            tem Estado ativa "active_lupa.png"
37        }
38    }
39 }

```

```

38
39
40 #Eventos:
41
42 Evento abrir_porta(1):
43 Se clique porta e porta está fechada e chave está em uso,
44 então porta muda para aberta e chave é removida.
45
46 Evento abrir_porta2(1):
47 Se clique porta2 e porta2 está fechada,
48 então porta2 muda para aberta.
49
50 Evento muda_cena():
51 Se clique porta2 e porta2 está aberta,
52 então muda para cena cena2.
53
54 Evento muda_cena2():
55 Se clique porta3 e porta3 está aberta,
56 então muda para cena cena.
57
58 Evento fim(1):
59 se clique porta e porta está aberta,
60 então fim de jogo.
61
62 Evento chave_inventario(1):
63 Se clique chave e cofre está aberto,
64 Então chave vai para o inventário.
65
66 Evento tentar_abrir_cofre():
67 Se clique cofre e cofre está fechado,
68 então pede código $codigo com mensagem "Qual é o código do cofre?" em (1000, 485) , se acertar
    faz abrir_cofre e se errar faz mostrar_erro.
69
70 Evento abrir_cofre(-):
71 cofre muda para aberto e chave muda para normal.
72
73 Evento mostrar_erro(-):
74 mostra mensagem "Código errado!" em (1000, 485).
75
76 Evento mensagem_trancada():
77 Se clique porta e porta está fechada e não chave está em uso,
78 então mostra mensagem "porta trancada! procura uma chave!" em (600,385) e toca porta-locked.
79
80 Evento arrasta_lupa():
81 Se clique lupa,
82 Então arrasta lupa, se deixar em cima de nota faz aumentar_nota e se não faz move_lupa.
83
84 Evento aumentar_nota(-):
85 nota_lupa muda para visivel.

```

```
86
87 Evento move_lupa(-):
88 lupa muda posição para (200,500).
89
90 Evento desaparecer_nota_lupa():
91 Se clique não nota_lupa e nota_lupa está visível,
92 então nota_lupa muda para null.
```

Listing A.4: Exemplo 3 de uma script na nossa DSL

