

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Modelagem da Progressão de Dificuldade em Jogos para Mapas Gerados Proceduralmente

Caio Vincenzo Reis Dima

JUIZ DE FORA
JANEIRO, 2023

Modelagem da Progressão de Dificuldade em Jogos para Mapas Gerados Proceduralmente

CAIO VINCENZO REIS DIMA

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Bacharelado em Sistemas de Informação

Orientador: Igor de Oliveira Knop

JUIZ DE FORA

JANEIRO, 2023

MODELAGEM DA PROGRESSÃO DE DIFICULDADE EM JOGOS PARA MAPAS GERADOS PROCEDURALMENTE

Caio Vincenzo Reis Dima

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM BACHARELADO EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

Igor de Oliveira Knop
D.Sc. Modelagem Computacional

Marcelo Caniato Renhe
D.Sc. Engenharia de Sistemas e Computação

Luciana Conceição Dias Campos
D.Sc. Engenharia Elétrica

JUIZ DE FORA
15 DE JANEIRO, 2023

Resumo

Jogos são uma atividade cultural que se transformou em uma indústria bilionária. O sucesso de um jogo está fortemente ligado na retenção da atenção do jogador, com base na motivação dos jogadores. Entre os modelos de motivações, o desafio e a maestria se destacam em jogos de progressão, como nos jogos de personificação (*role play games*), jogos de exploração de masmorras (*dungeon crawlers* ou mesmo os de morte permanente, como os *roguelikes*). Estes jogos mantêm o jogador na busca de melhorias para seu personagem, seja por equipamentos ou habilidades, sendo necessário a construção de algumas centenas de combinações para oferecer uma quantidade de conteúdo considerável para o jogador. Pequenos e grandes estúdios, exploram o conceito de geração procedural ao utilizar algoritmos especializados para gerar uma infinidade de mapas, itens ou desafios para os jogadores. Este trabalho explora um modelo de progressão de conteúdo para guiar a geração de inimigos e distribuição de atributos em mapas gerados proceduralmente. Um modelo de progressão é adaptado a um laboratório de geração procedural, novas heurísticas de distribuição de inimigos e recompensas são criados e sua influência no mapa final é avaliada sob a ótica do jogador. O ambiente de experimentação de código livre foi expandido com novas métricas e preparado para a adição de novos modelos para estudos subsequentes.

Palavras-chave: Desenvolvimento de jogos; geração procedural de conteúdo; sistemas de progressão.

Abstract

Games are a cultural activity that has become a billion-dollar industry. The success of a game is strongly linked to the retention of player attention, based on player motivation. Among the motivational models, challenge and mastery stand out in progression games, such as *role-playing games*, and dungeon exploration games (*dungeon crawlers* or even permanent death games, such as *roguelikes*). These games keep the player in the pursuit of improvements for his character, either by equipment or skills, requiring building a few hundred combinations to provide a considerable amount of content for the player. Small and large studios explore the concept of procedural generation by using specialized algorithms to generate a multitude of maps, items, or challenges for players. This paper explores a content progression model to guide enemy generation and attribute distribution in procedurally generated maps. A progression model is adapted to a procedural generation lab, new enemy and rewards distribution heuristics are created, and their influence on the final map is evaluated from the player's perspective. The open-source experimentation environment has been expanded with new metrics and prepared for the addition of new models for subsequent studies.

Keywords: game development; procedural content generation; progression systems.

Agradecimentos

Gostaria de agradecer principalmente aos meus pais, Dalci e Madalena, por terem me possibilitado trilhar o caminho que me levou onde estou hoje, por todo amor e sacrifícios, sempre me incentivando e ajudando como podiam.

Aos professores com os quais tive o privilégio de ter tido aula, por todo ensinamento e incentivo para continuar, além de toda visão e experiência que com certeza me tornaram o profissional que sou hoje.

Ao professor Igor Knop, por toda paciência de me explicar a mesma coisa diversas vezes, pela orientação e puxões de orelha.

Um agradecimento especial também à professora Luciana Campos, que desde meu primeiro ano me ajudou com todas as dúvidas que tive e também esteve presente em todos os projetos e bolsas que participei.

A todos os meus amigos, que me acompanharam durante essa incrível jornada, tornando meus dias mais divertidos, e que, mesmo com a vida nos afastando, levarei vocês sempre comigo. Em especial a Daide, Daniel e Victor, pelos incentivos e apoio emocional, nunca permitindo que eu desistisse dos meus sonhos.

Conteúdo

Lista de Figuras	6
1 Introdução	9
1.1 Apresentação	9
1.2 Contextualização	11
1.3 Descrição do Problema	12
1.4 Justificativa	12
1.5 Motivação	13
1.6 Questões	13
1.7 Objetivos	14
1.8 Organização do Documento	14
2 Fundamentação Teórica	15
2.1 Estudo dos jogos	15
2.2 Gêneros de jogos	15
2.3 <i>Roguelike</i>	16
2.4 Sistemas de Progressão	17
2.4.1 Progressão em jogos <i>idle</i>	19
2.5 Sistemas de Combate	22
2.6 Game Design	23
2.7 Geração procedural de conteúdo	24
2.8 Mapa de influência	24
2.9 PCGLab	25
3 Material e Métodos	28
3.1 Material	29
3.2 Trabalhos Relacionados	29
4 Desenvolvimento	30
4.1 Adaptação do Modelo de Progressão	30
4.1.1 Progressão dos Personagens	30
4.1.2 Progressão dos Inimigos	33
4.1.3 Progressão do Jogador	35
4.1.4 Progressão do poder nas salas e mapa geral	36
4.2 Posicionamento de elementos	38
4.2.1 Posicionamento de tesouros	39
4.2.2 Posicionamento de inimigos	40
4.3 Novas métricas sobre a geração	42
4.4 Avaliação da geração	44
4.4.1 Sala Pequena	44
4.4.2 Sala Média	46
4.4.3 Sala Grande	48
4.4.4 Conclusão da avaliação	50

5	Considerações Finais	54
5.1	Limitações e Trabalhos Futuros	55
	Bibliografia	57

Lista de Figuras

2.1	Diagrama parcial de gêneros, com destaque para o <i>roguelike</i> . Fonte: do autor, baseado em Adams (2014) e MobyGames (2021).	16
2.2	Captura de tela do jogo <i>Rogue</i> , de 1980. Fonte: SILVA (2015)	17
2.3	Dificuldade percebida de um jogo incremental. Fonte: Schreiber e Romero (2021)	19
2.4	Relação de custo de melhoria e aumento da taxa de retorno. Fonte: adaptado de King (2015a)	20
2.5	Relação de custo-benefício do jogo <i>Cookie Clicker</i> . Fonte: King (2015a)	21
2.6	Gráfico de custo-benefício do sistema de progressão do trabalho desenvolvido. Fonte: do autor.	22
2.7	Mapa de influência com decaimento. Fonte: captura de tela do <i>Procedural Content Generation Lab</i> (PCGLab), do autor.	25
2.8	Métricas de distâncias de elementos. Fonte: captura de tela do PCGLab, do autor.	26
2.9	Caminho entrada-saída e seu gráfico indicando as distâncias. O gráfico está normalizado e na legenda mostra quais os valores máximos de cada métrica: vermelho para inimigos; amarelo para tesouros e verde os teletransportes. Fonte: captura de tela do PCGLab, do autor.	27
4.1	Custo benefício dos atributos. Fonte: captura de tela do PCGLab, do autor.	33
4.2	Distribuição de atributos em inimigos. Fonte: captura de tela do PCGLab, do autor.	34
4.3	Distribuição do poder pelos atributos de inimigos. Fonte: captura de tela do PCGLab, do autor.	34
4.4	Interface com botões para aumentar atributos. Fonte: captura de tela do PCGLab, do autor.	35
4.5	Progressão de poder entre inimigos da mesma sala. Fonte: captura de tela do PCGLab, do autor.	37
4.6	Progressão de poder entre salas. Fonte: captura de tela do PCGLab, do autor.	38
4.7	Mapas de influência de posicionamento de tesouro, onde as células verdes são elegíveis para o posicionamento e as vermelhas não atendem aos critérios. Os tesouros estão circulados em amarelo para facilitar visualização. O caminho entrada-saída está destacado pelas setas vermelhas. Fonte: captura de tela do PCGLab, do autor.	41
4.8	Mapas de influência de posicionamento de inimigo, onde as células verdes são elegíveis para posicionamento, enquanto vermelhas não atendem aos critérios. Fonte: captura de tela do PCGLab, do autor.	42
4.9	Mapa de influência de poder dos inimigos. Fonte: captura de tela do PCGLab, do autor.	43
4.10	Gráfico com métricas de um caminho entrada-tesouro-saída. Fonte: captura de tela do PCGLab, do autor.	44
4.11	Mapa de influência do caminho entrada-saída. Fonte: captura de tela do PCGLab, do autor.	45

4.12	Mapa de influência do poder de tesouros Fonte: captura de tela do PCGLab, do autor.	46
4.13	Caminho da entrada até a saída do algoritmo de posicionamento antigo e novo, junto de seus gráficos, da sala pequena. Fonte: captura de tela do PCGLab, do autor.	47
4.14	Caminho que passa por todos os tesouros do algoritmo de posicionamento antigo e novo, junto de seus gráficos, da sala pequena. Fonte: captura de tela do PCGLab, do autor.	48
4.15	Caminho da entrada até a saída dos algoritmos de posicionamento antigo e novo, junto de seus gráficos, da sala média. Fonte: captura de tela do PCGLab, do autor.	49
4.16	Caminho que passa por todos os tesouros do algoritmo de posicionamento antigo e novo, junto de seus gráficos, da sala média. Fonte: captura de tela do PCGLab, do autor.	50
4.17	Caminho da entrada até a saída dos algoritmos de posicionamento antigo e novo, junto de seus gráficos, da sala grande. Fonte: captura de tela do PCGLab, do autor.	51
4.18	Caminho que passa por todos os tesouros dos algoritmos de posicionamento antigo e novo, junto de seus gráficos, da sala grande. Fonte: captura de tela do PCGLab, do autor.	52

Lista de Abreviações e Siglas

GPC Geração Procedural de Conteúdo. 11–13, 15, 17, 24, 25, 28, 38, 42, 54, 56

IA Inteligência Artificial. 24

PCGLab *Procedural Content Generation Lab*. 6, 7, 13, 25–30, 33–35, 37, 38, 41–52, 55

RPG *Role-Playing Game*. 10, 16, 17, 23, 24, 35

1 Introdução

1.1 Apresentação

No ano de 2020, a indústria de jogos apresentou um crescimento em relação ao ano anterior, mesmo com a pandemia de COVID-19, que prejudicou o desenvolvimento de diversos jogos (WIJMAN, 2021). Os sites especializados relatam crescimento entre 12% e 19,6% movimentando em torno de 160 bilhões de dólares e com expectativas de alcançar os 200 bilhões de dólares em 2023 (VIRGO, 2020; TAKAHASHI, 2021; BATCHELOR, 2020; WIJMAN, 2021).

O Brasil apresentou um crescimento de 5,1% no mesmo período, principalmente devido aos jogos de celular, chegando a um valor de mercado de 2,3 bilhões de dólares (ROUSSEAU, 2021). Com cada vez mais estúdios e profissionais atuando na área, brasileiros vêm se destacando na produção de jogos, como o jogo A Lenda do Herói, uma produção nacional com forte presença de elementos musicais (DUMATIVA; BROTHERS, 2016).

O interesse em investimento na indústria ocorre pelo constante crescimento que esse setor apresenta anualmente. Com os investimentos, há o avanço da tecnologia com a criação e aperfeiçoamento de novas ferramentas, e plataformas de desenvolvimento surgem para melhorar cada vez mais a qualidade dos jogos, e também permitir a criação de jogos mais complexos ou inovadores.

Além das pessoas estarem mais interessadas em jogar, houve também o aumento na quantidade de tempo gasto assistindo outras pessoas jogarem. Devido à pandemia e ao confinamento em casa, as pessoas buscaram outras maneiras de se entreter, e uma dessas formas foi assistindo conteúdos de jogos feitos por outras pessoas. O tempo gasto assistindo de forma online outras pessoas jogarem jogos eletrônicos está se aproximando do tempo gasto com jogos tradicionais, como futebol e vôlei (CHIOVATO, 2021).

Com um crescimento também em 2021, chegando a um total de 192 bilhões de dólares em receita, o ano de 2022 apresentou uma queda de 4,3%. O motivo seria

de reajuste dos anos de isolamento social que, apesar de terem sido bons para a venda de jogos, prejudicou a produção dos mesmos, fazendo com que os lançamentos de jogos fossem adiados. Apesar da queda, a soma das receitas dos anos de 2020 a 2022 são maiores que os projetados pré-pandemia (WIJMAN, 2022).

O estudo dos jogos é chamado ludologia, que engloba jogos de tabuleiro, de cartas e jogos eletrônicos. Existem diversas áreas de estudo, dentre elas o estudo do jogo em si, estudo do impacto social e cultural dos jogos, economias em jogos e, de especial interesse deste trabalho, o impacto na experiência dos jogadores e sua progressão em um jogo. Esse elemento é normalmente tratado na fase de *design*, muito preocupada com a experiência do jogador, planejada cuidadosamente pelo *game designer*. Mas aqui será estudado como os elementos de combate e geração procedural de conteúdo se interagem para este fim.

O *game design* é a aplicação de técnicas de *design* para elaborar mecanismos que compõem as regras e comportamentos dos modelos nos jogos e trabalhar na estética do jogo para atingir seu objetivo, podendo ser entretenimento, educacional, prática de exercício físico ou algum experimento. O desenvolvimento de jogos possui diversas etapas de produção, dentre elas: a concepção e imaginação do jogo; criação de níveis; *design* de personagens e balanceamento de dificuldade. Todas essas decisões de *game design* sofrem alteração conforme o tipo de jogo a ser desenvolvido e o público alvo definido para o produto.

Existem diversos tipos de jogos, que podem ser divididos em gêneros e subgêneros, que agrupam características em comum, principalmente em como o jogador interage com o jogo. No gênero *Role-Playing Game* (RPG), o jogador assume o papel de um ou mais personagens que se aprimoram ao vencer desafios, derrotando inimigos ou resolvendo quebra-cabeças. Nesse tipo de jogo, aprimorar o personagem é uma mecânica fundamental para vencer desafios cada vez mais difíceis, conforme o jogador progride no jogo. Em jogos de plataforma 2D, o jogo consiste basicamente em se movimentar e pular, desviando de obstáculos, enfrentando inimigos ou não, para chegar ao final da fase.

1.2 Contextualização

Existem três tipos de progressão em jogos: de conteúdo; de evolução do jogador; e de personagens. O primeiro deles é como o personagem avança através das fases até chegar ao final do jogo, ou ainda quando ele irá ganhar uma mecânica que permite ele acessar novas áreas. O progresso do jogador se refere à maneira em que ele aprende a jogar, descobrindo padrões de inimigos, memorizando comandos, sequências de golpes e novas mecânicas. Já o terceiro tipo de progresso é o aumento de força dos personagens, fazendo com que eles tenham mais vida e força de ataque, liberando novas habilidades e magias (BYCER, 2018). Este trabalho irá explorar principalmente o último tipo de progressão.

O foco do trabalho é no gênero de jogo *Roguelike*, que consiste em exploração de masmorras e batalhas contra inimigos. Sua principal característica é a progressão do personagem ao longo do jogo, adquirindo pontos de experiência, que o jogador acumula para passar de nível e conseguir novas habilidades e atributos, tornando-o mais forte para vencer novos desafios.

Uma técnica muito utilizada no desenvolvimento desse gênero de jogo é a Geração Procedural de Conteúdo (GPC), como mapa, inimigos, eventos e itens. A GPC é definida como a criação de elementos do jogo por meio de algoritmos, com pouca ou nenhuma influência do projetista (TOGELIUS et al., 2011a). Diferentes técnicas são aplicadas para diferentes objetivos, como mostrado em Togelius et al. (2011b).

Como cada jogo requer uma solução para um problema do escopo do gênero no qual ele se encaixa, o desenvolvimento pode ser facilitado com o uso de ferramentas para *game design*. Existem diferentes tipos de ferramentas, dentre elas as chamadas *game engines*, que fornecem um ambiente com diferentes recursos e bibliotecas para a criação do jogo, como, por exemplo, a Unity3D¹, que permite o desenvolvimento tanto de jogos 3D como 2D. Outro tipo de ferramenta são aquelas que não atuam diretamente na programação do jogo, mas ajudam o trabalho de *game design*, como, por exemplo, a ferramenta Machinations², onde é possível fazer a modelagem de sistemas complexos, como de economia ou de combate.

¹<https://unity.com/pt>

²<https://machinations.io/>

1.3 Descrição do Problema

Apesar de existirem bastantes ferramentas para auxiliar o trabalho de *game designers* no processo de construção de jogos, a maioria delas diz respeito a modelagem e programação, como nos casos das *game engine*. Existe, portanto, uma falta de ferramentas para ajudar no estudo e também na construção de sistemas internos de jogos, que muitas vezes ainda é realizado usando planilhas.

Esse cenário é ainda mais perceptível quando envolve GPC. Os diversos sistemas de um jogo, como de combate, financeiro ou progressão, precisam funcionar em conjunto, e existe um desafio ainda maior quando parte do conteúdo é gerado proceduralmente.

Em jogos do gênero *Roguelike*, uma ferramenta para analisar a progressão do personagem e a experiência do jogador com relação aos elementos gerados proceduralmente pode melhorar a experiência dos desenvolvedores desse tipo de jogo.

1.4 Justificativa

A indústria de jogos eletrônicos é uma das que mais cresce por ano. Somente no ano de 2020, houve um crescimento de 13% com relação ao ano anterior (TAKAHASHI, 2021). Apesar disso, desenvolver um jogo ainda é algo bastante custoso, e toda ferramenta para auxiliar essa tarefa é bem-vinda.

No cenário de jogos do gênero *Dungeon Crawler* e subgênero *Roguelike*, existe a tarefa de balancear os elementos do jogo para tornar a experiência do jogador agradável, apresentando um desafio enquanto lhe recompensa e deixa seu personagem mais forte. Essa etapa pode ser facilitada com uma ferramenta que mede o impacto de determinados elementos gerados proceduralmente, mostrando a influência que eles têm na experiência do jogador. Isso economizará tempo e recursos dos desenvolvedores, que terão uma melhor experiência no desenvolvimento.

1.5 Motivação

Um dos motivos é o avanço no conhecimento acerca da GPC no ambiente de estudo PCGLab, um trabalho em andamento, desenvolvido por Costa (2020) com contribuições de (SANTANA, 2021) e (VERDE, 2021), cujo objetivo é estudar o impacto da GPC na experiência do jogador. Essa ferramenta ainda não conta com um modelo de progressão associado à criação e posicionamento dos elementos gerados, um elemento muito importante em jogos.

Adicionalmente, é do interesse do autor continuar no estudo de jogos e, desenvolver novas ferramentas capazes de auxiliar o *design* de jogos é uma forma de se aprofundar na literatura e ainda ajudar outros desenvolvedores a entender o impacto da geração procedural na experiência e progresso do jogador.

1.6 Questões

Em um jogo do gênero *Dungeon Crawler*, diversas técnicas de GPC são utilizadas, tanto para a geração do mapa como para geração de itens e encontros. Dependendo de como esses elementos são gerados, a experiência do jogador pode ser afetada. Com base nisso, a pesquisa visa responder à questão geral “Como a geração procedural de conteúdo afeta a experiência do jogador?”.

Além disso, também é necessário considerar como será a progressão de dificuldade. Esse elemento do jogo é de extrema importância por ele ser responsável por fazer o jogador se sentir imerso com o desafio enfrentado. No contexto do trabalho, será por meio do combate que a dificuldade será modelada. Logo, como deve ser feita a modelagem e qual deve ser o nível de dificuldade do combate, para que a pesquisa responda à questão “Como o sistema de progressão de dificuldade afeta a experiência do jogador?”.

Outro fator importante a se considerar é a progressão do personagem do jogo. O sistema de recompensas com itens e pontos de atributos precisam dar o sentimento de progressão ao jogador. Isso precisa se conciliar com a dificuldade para que vencer os inimigos não seja nem tão fácil, nem tão difícil. Nesse caso, a questão a ser respondida é “Como a progressão afeta a experiência do jogador?”. O trabalho deve evidenciar modelos

de progressão que considerem os diversos elementos do jogo.

1.7 Objetivos

O objetivo principal deste trabalho é apresentar um método de avaliação de modelos de progressão aliado à geração procedural de conteúdo. Esse método será implementado em uma ferramenta que permitirá ao *game designer* modelar o impacto da geração procedural nos sistemas de combate e na progressão do jogador, ou seja, como os desafios e encontros gerados proceduralmente têm impacto na experiência do jogador. Para atingir esse objetivo geral, os seguintes objetivos específicos são necessários:

- Criar um modelo em alto nível para sistemas de progressão de dificuldade utilizando o combate, simulando o desafio para os jogadores;
- Criar um modelo de progressão do jogador, para simular o quanto ele consegue passar pelos desafios;
- Ligar os modelos em um sistema de geração procedural de conteúdo, para vincular o sistema de progressão;
- Avaliar a geração de conteúdo sob a ótica do jogador, realizando medições e aplicando formulários de aceitação.

1.8 Organização do Documento

Este documento está organizado em cinco capítulos. Além desta Introdução, o Capítulo 2 faz uma fundamentação dos principais conceitos necessários para o entendimento deste trabalho. O Capítulo 3 apresenta o método aplicado para a execução do trabalho e o Capítulo 4 mostra todo desenvolvimento acerca do problema abordado. Já o Capítulo 5 contém as considerações finais, assim como as limitações enfrentadas e sugestões para trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo serão apresentados os principais conceitos utilizados para execução do trabalho. Num primeiro momento, uma revisão do estudo e indústria de jogos é realizada e, na sequência, os fundamentos de *game design*, gênero de jogos e o uso da GPC. Por fim, o conceito de mapas de influência será revisto, pois são utilizados tanto na geração quanto na avaliação dos resultados.

2.1 Estudo dos jogos

O estudo dos jogos, também chamado de ludologia, possui diversos objetos de estudo, desde jogos de tabuleiro, esportes, atividades lúdicas, mas a de principal interesse desse trabalho são os jogos eletrônicos. Diferentes áreas de pesquisas abordam o tema de jogos de diferentes maneiras com objetivos diversos.

Dentre elas, existe o estudo voltado à engenharia, específica de jogos eletrônicos, cujo objetivo é entender e desenvolver novas técnicas de som, gráficos mais realistas, aplicações de inteligência artificial ou melhora de desempenho. Outra área de pesquisa são as ciências sociais, que tenta entender qual o impacto dos jogos na sociedade e como eles afetam o jogador. A área de interesse do trabalho é a de *game design*, que visa construir um jogo da melhor maneira para que a experiência do jogador seja a melhor possível, elaborando e juntando diferentes sistemas num jogo para funcionarem em conjunto.

2.2 Gêneros de jogos

Assim como filmes e livros, os jogos podem ser classificados em gêneros e subgêneros, podendo inclusive pertencer a mais de um (APPERLEY, 2006). Porém, eles não devem ter como base a estética ou narrativa para sua categorização. Jogos possuem como principal característica a interação e participação do jogador, e é através delas que são classificados (WOLF, 2008).

Existe certa dificuldade ao classificar jogos, principalmente pelas diferentes opiniões de autores e pela mistura de elementos de subgêneros diferentes em jogos. Em Adams (2014) são definidos alguns dos principais gêneros de jogos, divididos em diversos subgêneros, cada um com um conjunto de características que o diferencia dos outros. Além dele, a base de dados MobyGames possui um glossário explicando a classificação de jogos a partir de diferentes perspectivas (MOBYGAMES, 2021).

O subgênero *roguelike*, que possui elementos do gênero de ação, aventura e RPG, é de particular interesse deste trabalho e está detalhado na Seção 2.3. A Figura 2.1 mostra um recorte de um diagrama de gêneros de jogos, com a hierarquia do subgênero que foi trabalhado.



Figura 2.1: Diagrama parcial de gêneros, com destaque para o *roguelike*. Fonte: do autor, baseado em Adams (2014) e MobyGames (2021).

2.3 *Roguelike*

Roguelike é um subgênero de jogo eletrônico cuja origem vem do jogo *Rogue* (KUITTINEN, 2001), que inspirou diversos jogos e, junto de outros, motivou a criação do gênero (ZAPATA, 2017). Sua definição ainda está em discussão (MOSS, 2020), mas em 2008 foi realizada uma tentativa de definir as principais características do gênero, dentre elas: a morte permanente do personagem; ações em turno; e geração procedural de conteúdo (principalmente de mapa e itens). Historicamente, a estética visual é por gráficos em terminais em modo texto puro (Figura 2.2). É importante ressaltar que a ausência de

algumas dessas características não exclui o jogo do gênero, nem a presença garante que ele faça parte do mesmo.

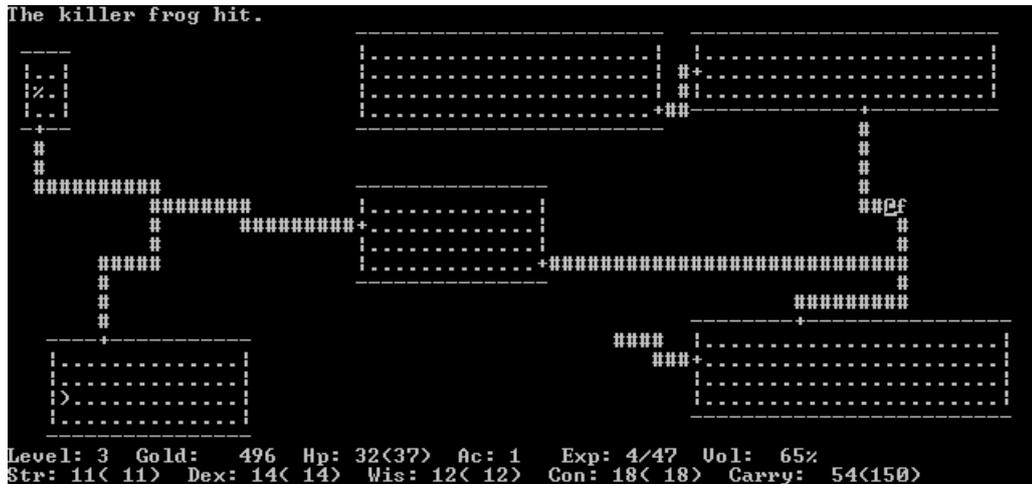


Figura 2.2: Captura de tela do jogo *Rogue*, de 1980. Fonte: SILVA (2015)

A partir dos anos 2000, diversos jogos de outros gêneros começaram a incorporar elementos de *roguelike*, dando origem ao termo “rogue-lite”, ou ainda “roguelike-like” (WINKIE, 2021). Esses jogos possuem algumas características do *roguelike*, como morte permanente e GPC, mas outras deixam de lado, principalmente as de interface baseada em caracteres e ações turno a turno.

O jogo utilizado neste trabalho, considerando as explicações dadas anteriormente, é considerado um “rogue-lite” por possuir algumas características do *roguelike* e também do gênero de RPG de ação 2D. Além disso, o jogo também apresenta características de *Dungeon Crawl*, outro subgênero do RPG que consiste em percorrer um labirinto ou masmorra gerado proceduralmente, enquanto enfrenta inimigos, soluciona quebra-cabeças e coleciona tesouros (BREWER, 2016).

2.4 Sistemas de Progressão

Progressão é um dos principais elementos de *game design*, uma vez que é ela que define o ritmo do jogo e é responsável por boa parte da experiência do jogador. Ela pode se referir a diferentes elementos como: a velocidade que o personagem ganha poder; a introdução de novos mecanismos e elementos ou; o ritmo de contar a história. Um sistema de progressão

de um jogo complexo, com diferentes habilidades e subsistemas, deve pensar em como introduzir esses elementos gradualmente aos jogadores.

De acordo com Dormans (2013), a progressão em um jogo deve ser cuidadosamente pensada a fim de não sobrecarregar o jogador. Em muitos jogos, até mesmo nos mais simples, é comum encontrar um excesso de regras, estratégias e elementos visuais, que se introduzidos todos juntos, logo no início, pode fazer com que o jogador perca a motivação.

Uma maneira de evitar essa sobrecarga é introduzir novas mecânicas e elementos através de tutoriais, no formato de níveis introdutórios ou por meio de pequenos trechos durante o jogo (DORMANS, 2013). Um jogo que faz isso muito bem é *God of War*, que a cada novo poder introduzido, exige que o jogador o utilize para avançar.

A progressão exige que exista algum tipo de desafio que o jogador deva superar, pois caso contrário ele conseguiria atingir o objetivo e receber as recompensas do jogo sem dificuldade alguma (SCHREIBER; ROMERO, 2021). Em um jogo, diferentes desafios são criados e cada um deles com diferentes níveis de dificuldade. A dificuldade percebida pelo jogador, que muitas vezes é maior do que a dificuldade real do desafio, pode ser calculada com a combinação de quatro elementos: nível de habilidade do jogador; nível de poder do personagem do jogador; nível de poder dos inimigos e; desafio de habilidade (SCHREIBER; ROMERO, 2021).

O desafio enfrentado pelo jogador é essencial para que ele se sinta imerso no jogo. Uma das teorias da psicologia que estuda a imersão de uma pessoa em uma atividade é a teoria do *flow* (CSIKSZENTMIHALYI et al., 1990 apud SCHREIBER; ROMERO, 2021). O estado de *flow* é atingido no equilíbrio entre o nível do desafio e da habilidade de quem executa a tarefa. Schreiber e Romero (2021) destacam que um jogo pode se tornar entediante caso não aumente a dificuldade de seus desafios, pois o jogador, ao longo do tempo, melhora seu nível de habilidade. Logo, sem aumento de dificuldade, o jogo tende a ter uma dificuldade percebida reduzida a ponto de ser fácil demais. Cabe, então, ao *game designer* ou ao sistema de geração de conteúdo prover esse incremento constante na dificuldade.

2.4.1 Progressão em jogos *idle*

Em jogos muito complexos, a progressão pode ser de difícil identificação em função dos diferentes sistemas e elementos estéticos. Mas um gênero de jogo leva isso ao fundamental, os chamados “idle games”, também conhecidos como jogos incrementais. Esse tipo de jogo requer mínima interação do jogador e utiliza técnicas simples de recompensa para atrair jogadores (KING, 2015b).

O desafio percebido desse tipo de jogo é muito influenciado pelo crescente nível de poder, que no caso é a renda crescente, e os desafios são os custos, que também aumentam ao longo do tempo, para obter novas melhorias. Além disso, o nível de poder do jogador, ou seja, a quantidade de pontos ou dinheiro gastos, também é alta. Porém, como explicado em Schreiber e Romero (2021), o crescimento de habilidade do jogador é muito limitado, uma vez que o jogo não exige que o jogador seja habilidoso para jogá-lo, enquanto o nível de desafio de habilidade é constante, pois não apresenta novas mecânicas.

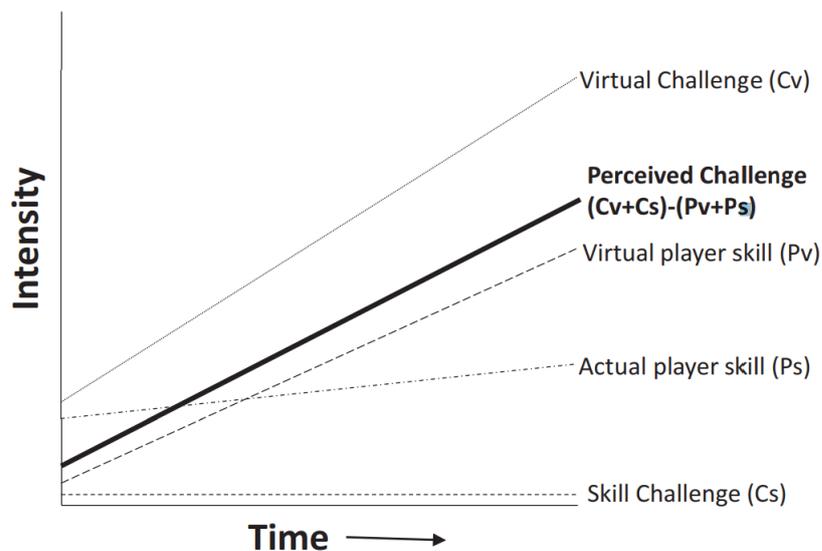


Figura 2.3: Dificuldade percebida de um jogo incremental. Fonte: Schreiber e Romero (2021)

Jogos incrementais possuem diversos recursos que o jogador pode obter e aprimorar para ganhar ainda mais pontos. É possível obter a relação de custo-benefício de cada um desses recursos, uma vez que cada um tem uma taxa de retorno e um custo associado. O crescimento exponencial do custo de melhoria de um recurso é interessante para haver uma diversificação entre quais recursos o jogador irá melhorar. Em King (2015a)

foi feita uma análise do custo benefício do jogo *Clicker Heroes*³, da primeira melhoria disponível, representado por heróis no contexto do jogo. Para esse herói, a equação de custo é dada pela Equação 2.1, e o retorno é linear, com um aumento de 5 para cada ponto de melhoria. Nela, o custo para a próxima compra $C(n)$ é dado em função exponencial, do produto do custo base 50 com uma base arbitrária de expoente n , em que n significa a quantidade de pontos daquele herói que o jogador possui no momento. A Figura 2.4 mostra a curva de custo exponencial e a curva de retorno linear.

$$C(n) = 50 * 1,07^n \quad (2.1)$$

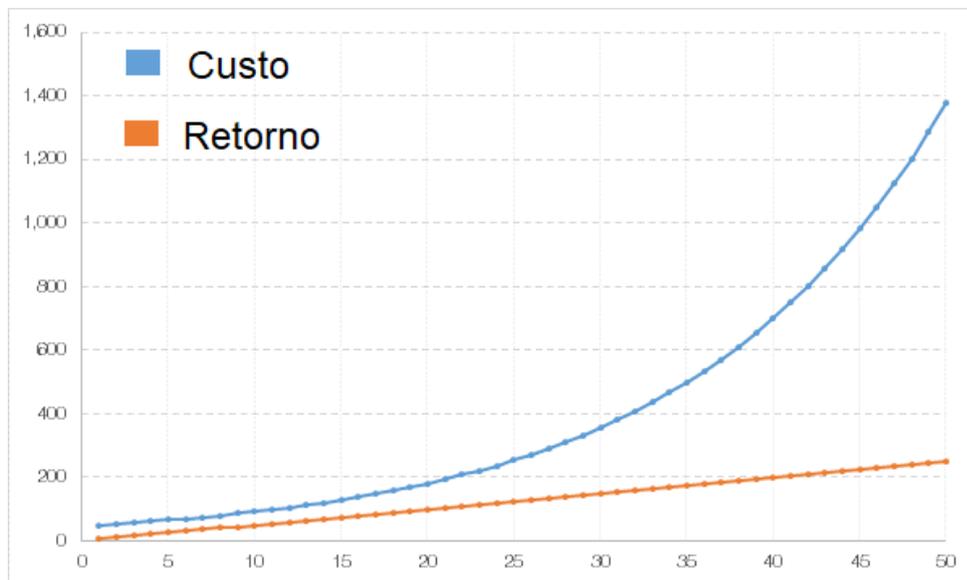


Figura 2.4: Relação de custo de melhoria e aumento da taxa de retorno. Fonte: adaptado de King (2015a)

Ainda em King (2015a) foi feita uma análise do custo benefício do jogo *Cookie Clicker*⁴, para cada tipo de melhoria disponível. A equação é parecida com o jogo *Clicker Heroes*, com a diferença da base. Nesse jogo, cada recurso possui um custo base e retorno diferente. A equação do custo de um recurso é dada pela Equação 2.2.

$$C(n) = C_0 * 1,15^n \quad (2.2)$$

A Figura 2.5 ilustra como os recursos do jogo *Cookie Clicker* conduzem a

³<https://www.clickerheroes.com/>

⁴<https://orteil.dashnet.org/cookieclicker/>

progressão do jogador. As curvas são paramétricas em função das primeiras 200 melhorias e cada curva representa uma melhoria diferente do jogo. Ambos eixos estão em escala logarítmica, o eixo y representa o custo de melhoria e o eixo x representa o retorno. Ao fazer o custo de melhoria ficar mais alto do que o custo para um próximo elemento, os jogadores buscam otimizar suas ações com base no custo-benefício percebido ou calculado automaticamente pelo jogo: uma melhoria se torna menos eficiente quando outra a cruza.

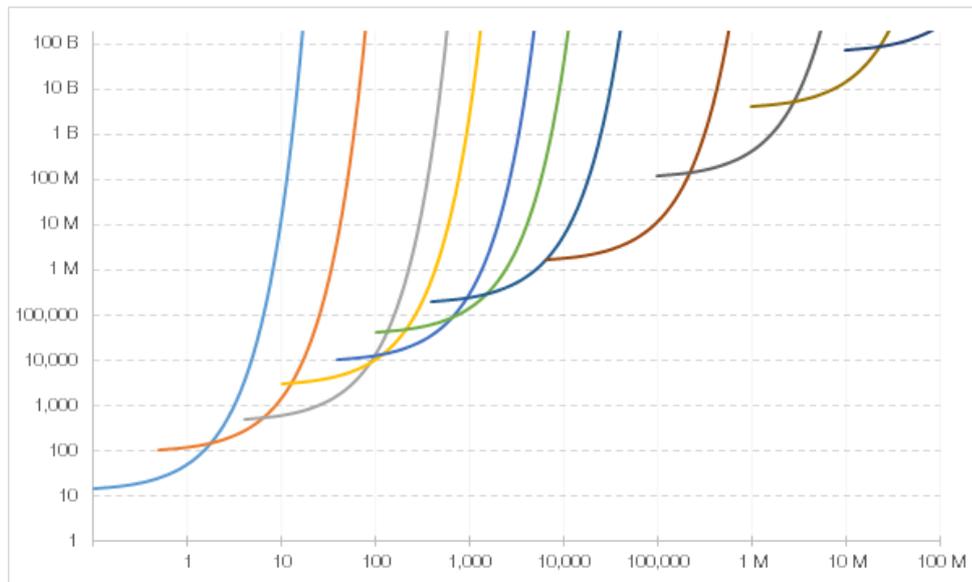
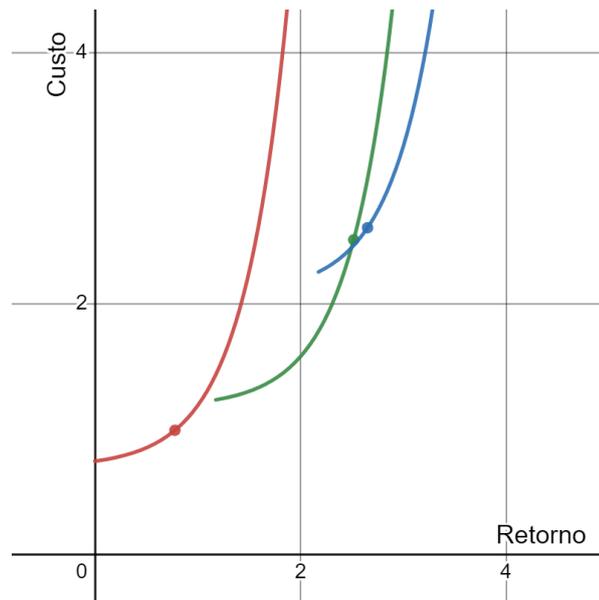


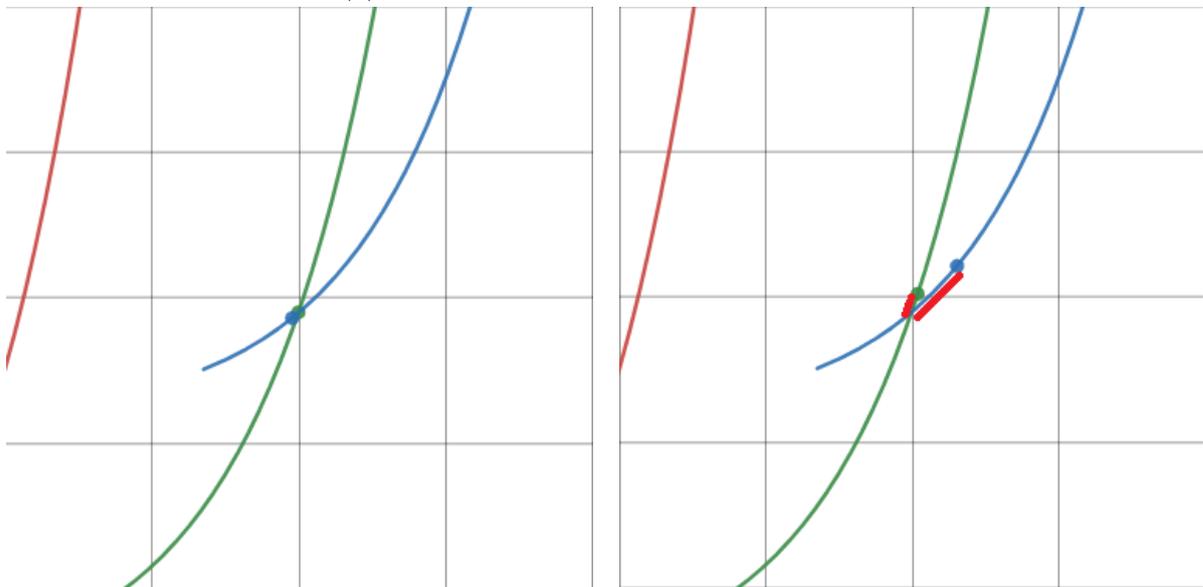
Figura 2.5: Relação de custo-benefício do jogo *Cookie Clicker*. Fonte: King (2015a)

Uma maneira de entender esse gráfico é com o exemplo da Figura 2.6, que representa o custo-benefício do sistema de progressão do trabalho desenvolvido, explicado na Seção 4.1. A Figura 2.6b e Figura 2.6c são uma ampliação da Figura 2.6a, representados pelo custo no eixo vertical e o retorno no eixo horizontal. Na Figura 2.6b, é mostrado que as melhorias da curva azul e verde têm praticamente o mesmo custo, mas ao adquirir uma melhoria de cada, como mostrado na Figura 2.6c e destacado pelas linhas em vermelho, o retorno da melhoria azul é muito maior do que a da verde.

O modelo de progressão de jogos incrementais se mostra um ótimo ponto de partida para ser integrado neste trabalho, por ser o modelo mais fundamental. A partir dele, sua influência pode ser adicionada na geração procedural e modelos mais complexos podem evoluir posteriormente. A Seção 4.1 apresenta esse esforço.



(a) Gráfico de custo-benefício do sistema.



(b) Custos iguais para as curvas azul e verde.

(c) Um ponto a mais para cada melhoria.

Figura 2.6: Gráfico de custo-benefício do sistema de progressão do trabalho desenvolvido.
Fonte: do autor.

2.5 Sistemas de Combate

Um dos principais elementos dos jogos eletrônicos de diversos gêneros é o combate. Existe uma grande variedade de sistemas de batalha dentre os diversos tipos de jogos. Osborn et al. (2018) tenta definir quais elementos um jogo precisa ter para que se possa dizer que ele possui um sistema de combate.

O artigo utiliza como ferramenta avaliativa a lógica operacional, a qual é a relação entre a representação gráfica de algo, ou seja, a maneira como é representada,

e como é feita a implementação por algoritmos. Os autores afirmam que para um jogo possuir um modelo de combate é necessário que ele deixe explícito que uma batalha está acontecendo entre os envolvidos, os mesmos precisam realizar ações violentas entre eles e o conflito precisa chegar a um fim, seja devido à derrota ou ao tempo.

Os combates podem ser por turno ou em tempo real, podendo ter variações. Em batalhas por turno, cada um dos personagens envolvidos executa sua ação em turnos alternados, seguindo uma ordem. Depois que todos realizarem suas ações, um novo turno se inicia, mantendo a ordem.

Em jogos com combate em tempo real não existem turnos, o tempo se passa de forma contínua e os personagens executam suas ações de maneira concorrente. A consequência de cada ação é percebida imediatamente por todos os envolvidos, adicionando um desafio de destreza ao jogador.

O combate em jogos do estilo RPG sofrem influência direta de atributos que os personagens possuem, como vida máxima, dano e velocidade de movimento. Schreiber e Romero (2021) explicam que todos esses atributos se resumem a manter o personagem vivo, seja por aumentar sua resistência a dano, dar mais dano para encurtar o tempo de combate ou aumentar a taxa de evasão para se evitar levar dano. Aumentar esses atributos faz com que o personagem fique mais efetivo em um combate, permitindo que ele participe de mais encontros e cada vez mais difíceis, geralmente relacionados a recompensas melhores.

2.6 Game Design

Game Design é o processo iterativo de criar conteúdo e regras para o jogo, considerando a estética e os mecanismos do jogo. Segundo Brathwaite e Schreiber (2008), um bom *game design* deve: ter como foco o jogador; criar objetivos interessantes o suficiente para motivá-lo e; criar regras não para obrigar o jogador a fazer algo, mas que ele deve seguir enquanto toma suas próprias decisões, com impacto direto no resultado do jogo.

Com o aumento de complexidade dos jogos e seus diversos sistemas, houve necessidade de especialização dos *game designers*, levando a criação de diversos tipos de funções em uma equipe. Dentre elas, se destacam o *level design*, que consiste em

construir as fases do jogo, responsável pelo formato do mapa e posicionamento de objetos e inimigos. Outro tipo de função de *game design* que se destaca é o *systems design*, que cria os modelos matemáticos de diferentes sistemas do jogo, como economia, sistema de combate e a progressão do jogo. Entretanto, ambos devem atuar em conjunto para que a experiência final para o jogador seja significativa.

2.7 Geração procedural de conteúdo

Em desenvolvimento de jogos, uma forma de economizar tempo e dinheiro, além de possibilitar uma maior variabilidade de cenários e elementos de jogo, é a técnica de Geração Procedural de Conteúdo (GPC). Esse método é a criação de conteúdo, como mapa, inimigos, itens e encontros, realizada por algoritmos com pouca interação humana (TOGELIUS et al., 2011a)

Existem diversas aplicações em jogos digitais que se pode utilizar GPC. Nos primeiros jogos do gênero *roguelike* a técnica era utilizada para criar o mapa, formando as salas e posicionando inimigos, passagens e objetos. Atualmente, GPC é bastante utilizada em jogos de RPG em sistemas de recompensa e saque de itens. Um exemplo é a série *Borderlands*, que cria uma grande variedade de armas com diversos atributos diferentes, com base no nível do personagem.

2.8 Mapa de influência

Mapas de influência são muito utilizados como técnica de tomada de decisão estratégica de um ou mais agentes. Eles consistem em uma representação espacial do conhecimento que um agente controlado por uma Inteligência Artificial (IA) possui (TOZOUR, 2001, p. 287). Sua representação se dá através da divisão de um espaço em diversas células, em formato de grade, atribuindo valores a cada uma delas. Esses valores podem representar diferentes tipos de informações, desde força, distâncias, perigo ou qualquer outra métrica que pode ser medida (MARK, 2019).

Tozour (2001) explica que o valor de uma célula tem seu valor propagado para um certo número de células vizinhas. A quantidade de vezes que o valor será propagado

depende da função de propagação e da taxa de decaimento. Quando duas entidades estão próximas e suas influências colidem, elas são somadas.

A Figura 2.7 exemplifica um mapa de influência com origem em um tesouro, com uma função de decaimento representada pela Equação 2.3, com valor inicial de 100, decaimento de 15% e n é a distância da célula de origem.

$$V_n = 100 * 0.85^n \quad (2.3)$$

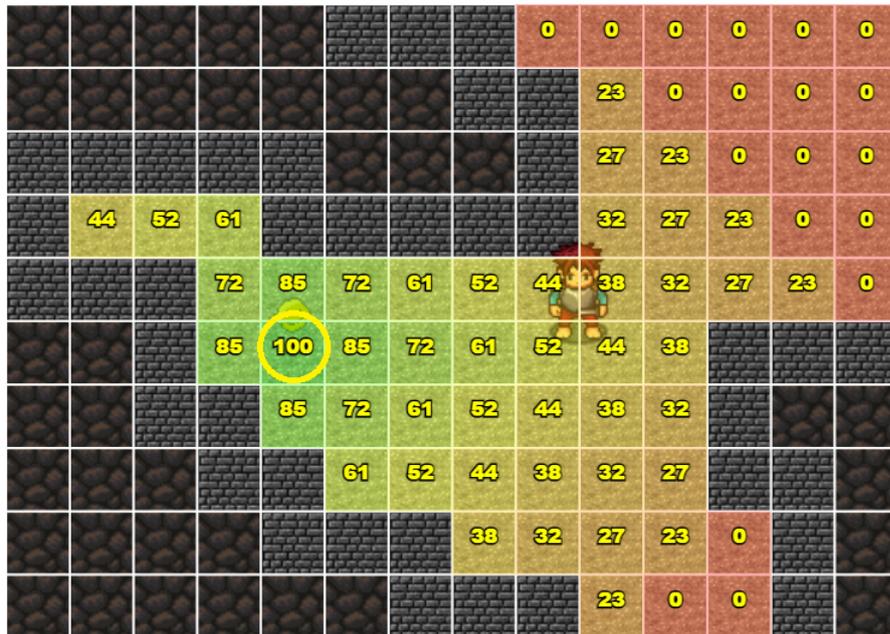


Figura 2.7: Mapa de influência com decaimento. Fonte: captura de tela do PCGLab, do autor.

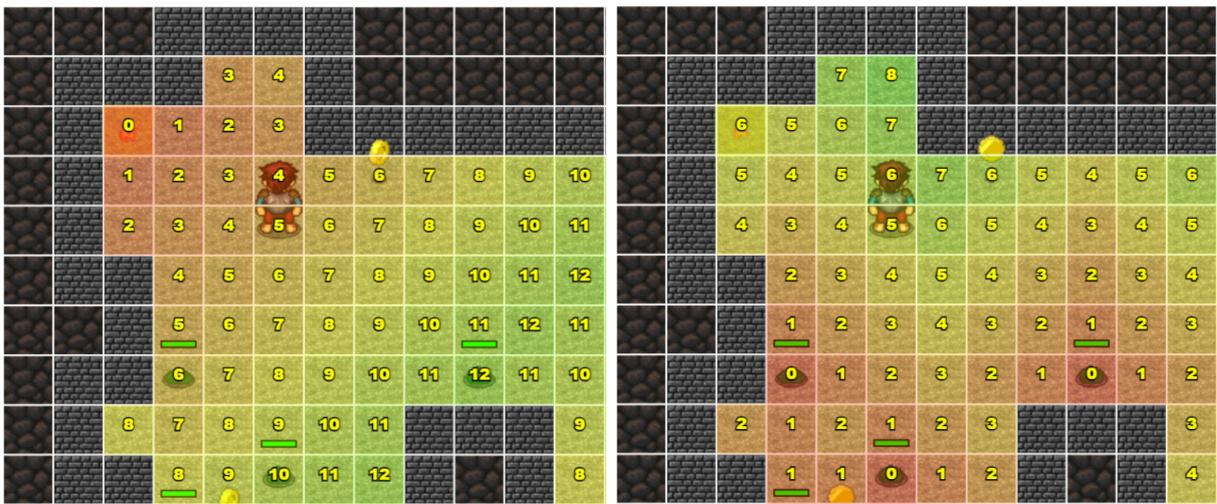
A propagação da influência fornece informações relevantes para a tomada de decisão, porém muitas vezes um mapa sozinho não é o suficiente para situações complexas que envolvem muitas variáveis. A combinação de diferentes mapas de influência pode ser mais útil para a decisão, a depender do que se deseja fazer. Geralmente, a melhor maneira de se combinar mapas é utilizando soma ponderada dos valores (TOZOUR, 2001).

2.9 PCGLab

O trabalho foi inserido no ambiente de estudo PCGLab, desenvolvido por Costa (2020) como uma tentativa de entender como a GPC aplicada à geração de mapas afeta a ex-

perícia do jogador em um jogo *dungeon crawler*. A ferramenta, apesar de estar nos estágios iniciais e apresentar algumas limitações, é jogável e apresenta diferentes métricas para avaliação da experiência do jogador (SANTANA, 2021). A geração do mapa é realizada proceduralmente utilizando as técnicas de autômatos celulares (COSTA, 2020) e *Wave Function Collapse* (VERDE, 2021), aplicando heurísticas para o posicionamento dos elementos, utilizando mapas de influência e outras métricas.

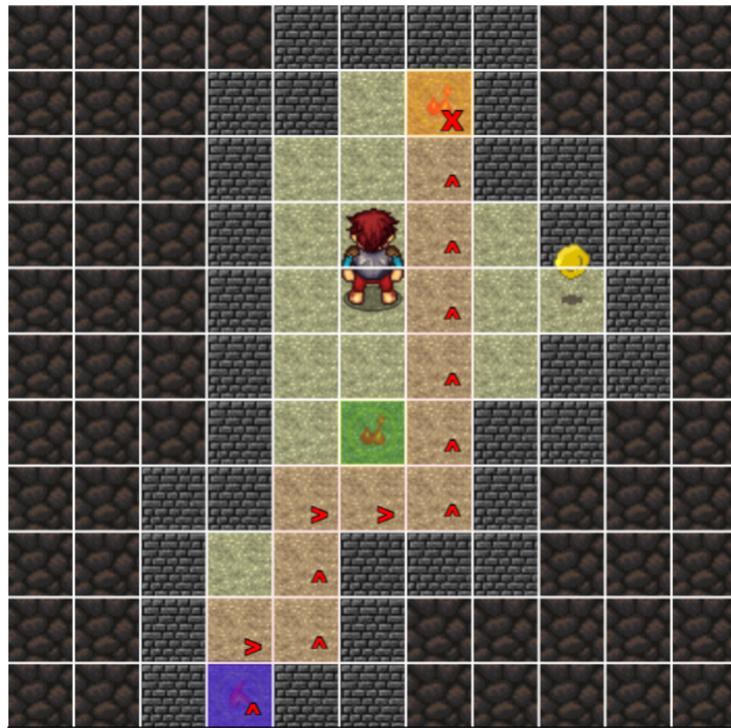
Além do mapa, o jogo é formado por tesouros, inimigos e fogueiras, e suas salas são conectadas por teletransportes, permitindo ao jogador navegar entre elas. Os inimigos servem como desafio para o jogador, engajando em combate e perseguindo o jogador quando o mesmo chega a pelo menos dois blocos de distância dele (conhecido pelo termo raio de agressividade ou *aggro radius*). Os tesouros sevem de pontuação no final do jogo e as fogueiras, também chamadas de *firezones*, servem para reiniciar um tempo decrescente que, ao chegar em zero, faz com que o jogador perca. Cada um desses elementos fornece métricas, sejam elas simples ou compostas, mostradas na Figura 2.8. Além disso, a ferramenta possui caminhos entrada-saída, entrada-tesouros-saída e o caminho tomado pelo jogador em tempo real, exemplificado na Figura 2.9, que permitem analisar o posicionamento desses elementos a partir de suas distâncias para cada um desses caminhos. Quando as linhas de distâncias encostam o eixo dos passos, temos a informação que o elemento está presente em um bloco do caminho.



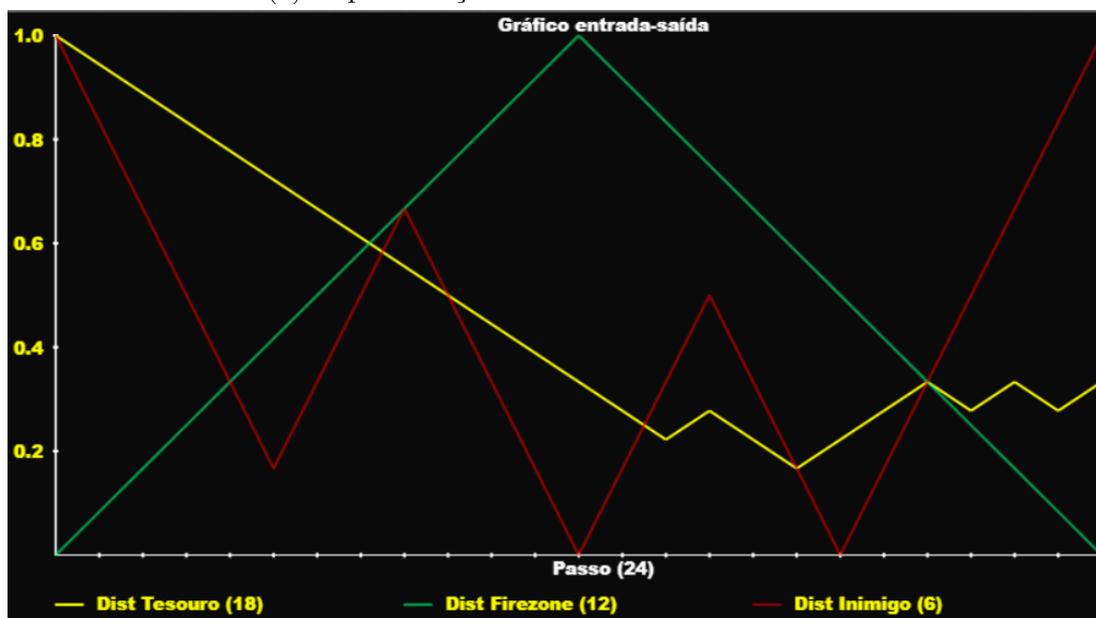
(a) Distância de teletransportes.

(b) Distância de inimigos.

Figura 2.8: Métricas de distâncias de elementos. Fonte: captura de tela do PCGLab, do autor.



(a) Representação do caminho entrada-saída.



(b) Gráfico do caminho entrada-saída.

Figura 2.9: Caminho entrada-saída e seu gráfico indicando as distâncias. O gráfico está normalizado e na legenda mostra quais os valores máximos de cada métrica: vermelho para inimigos; amarelo para tesouros e verde os teletransportes. Fonte: captura de tela do PCGLab, do autor.

3 Material e Métodos

A presente pesquisa exploratória, de caráter qualitativo, busca um maior entendimento de como alinhar a progressão de dificuldade do jogo e o modelo de progressão de jogador com a geração procedural de conteúdo. Em cima dos modelos são realizadas medições, e o resultado é analisado do ponto de vista da experiência do jogador. Apesar de a mesma ser algo subjetivo, com os modelos e medidas foi criado um ferramental no qual estes possam ser refinados e comparados com relatos de pessoas reais.

Como subproduto do trabalho, foi criada uma ferramenta para auxiliar o processo de *game design*, principalmente na tomada de decisão em relação ao sistema de progressão, avaliando o impacto da GPC sobre a experiência do jogador.

O trabalho é faseado, sendo a primeira delas o desenvolvimento de um modelo para os inimigos, no qual foram escolhidas as variáveis que o compõem. Em seguida foi criado um modelo de progressão que foi medido através do combate do jogador com os inimigos. Na sequência, essas variáveis formaram uma nova heurística de posicionamento de elementos nos mapas gerados e as medidas foram observadas e comparadas com os trabalhos anteriores.

Tanto o sistema de combate quanto de progressão foram implementados em um ambiente de experimentação de GPC, o PCGLab, alterando o posicionamento de inimigos e tesouros pelo mapa, permitindo realizar diversos testes e análises de impacto de alteração dos parâmetros. Os dados gerados pelo sistema foram comparados ao posicionamento do algoritmo antigo, analisando o caminho que passa por todos os tesouros e o caminho mínimo da entrada até a saída.

3.1 Material

Para o desenvolvimento foi utilizado o editor de código-fonte Visual Studio Code ⁵ com a linguagem de programação Javascript⁶. Para o controle de versão foi utilizado o Github⁷. Todo desenvolvimento foi feito no computador pessoal do autor, portanto não é necessário nenhum servidor para executar o jogo. O repositório do código pode ser acessado por <https://github.com/ufjf-gamelab/pcglab>.

3.2 Trabalhos Relacionados

O trabalho é uma continuação do desenvolvimento da ferramenta PCGLab, desenvolvida por Costa (2020), com a contribuição direta de SANTANA (2021) e VERDE (2021). Todos os três trabalhos exploram a geração procedural através da construção do mapa, além de avaliar o posicionamento dos elementos.

⁵<https://code.visualstudio.com/>

⁶<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

⁷<https://github.com/>

4 Desenvolvimento

Este capítulo explica o desenvolvimento das novas funcionalidades no PCGLab, descrevendo como foi feita a adaptação do modelo de progressão e suas equações, passando também pelo novo algoritmo de geração e posicionamento de elementos nos mapas. Por fim, descreve também a avaliação dos modelos em relação aos resultados de trabalhos anteriores.

4.1 Adaptação do Modelo de Progressão

O sistema de progressão adaptado para o PCGLab é baseado no modelo mais simples dos jogos incrementais, utilizando variações de sua função exponencial de custo. Ao invés de um valor monetário como recurso, há o acúmulo de “poder” dos personagens como variável principal. Essa variável é utilizada para se medir quão forte é um personagem, sendo recurso necessário para aumentar atributos, que impactam diretamente no resultado do combate. Como é a variável principal que descreve a progressão, ela também é utilizada em um novo algoritmo de posicionamento dos inimigos e tesouros, conforme pode ser visto na Subseção 4.1.4.

Tanto inimigos quanto o personagem do jogador são tratados como personagens do jogo, e compartilham da mesma modelagem de atributos que influenciam na jogabilidade, além de também seguirem o mesmo modelo de progressão de custo de poder. Nas subseções a seguir, é detalhado como a adaptação do modelo foi realizada.

4.1.1 Progressão dos Personagens

A fim de simplificar o modelo, foram utilizados três atributos, sendo eles vida máxima, força de ataque e velocidade de movimento. O princípio básico é que quanto maiores são os valores desses atributos, mais forte é o personagem. A vida de um personagem significa a quantidade de dano que ele pode sofrer antes de morrer, força de ataque é a quantidade de pontos de vida que é retirado do alvo ao atacar e velocidade é o quão

rápido um personagem consegue se mover.

Assim como em jogos incrementais, o sistema de progressão desenvolvido possui um custo, em poder, para aumentar os atributos de um personagem. Quanto mais poder ele tem, mais ele consegue aumentar seus atributos. O sistema utiliza uma curva de custo dada pela Equação 4.1, onde C_n é o custo do n ésimo ponto de um atributo, C_0 é o custo base e G_b é a constante associada ao crescimento de custo de cada um dos atributos.

$$C(n) = C_0 * G_b^n \quad (4.1)$$

O sistema da Equação 4.2 mostra o custo de cada um dos atributos, com suas respectivas constantes e custos iniciais. $H(n)$ representa o custo da vida máxima, $A(n)$ o custo do ataque e $V(n)$ o custo da velocidade para o n ésimo ponto de cada atributo. Para cada equação, n representa a quantidade de pontos do respectivo atributo.

$$C_H(n) = 15 * 1.15^n \quad (4.2a)$$

$$C_A(n) = 5 * 1.12^n \quad (4.2b)$$

$$C_V(n) = 120 * 1.5^n \quad (4.2c)$$

A escolha das constantes foram definidas de forma empírica, visando controlar a frequência de se obter atributos através do seu crescimento de custo. Dessa forma, o atributo velocidade, por ter um impacto no combate maior que os outros, possui um taxa e custo inicial maiores.

Além dos atributos, todo personagem possui um poder total, que consiste no somatório do custo de todos os pontos de cada atributo que ele possui, mostrado em Equação 4.3 representando por P . Nela, a representa a quantidade total de pontos no atributo vida máxima que o personagem tem, da mesma forma que b representa a quantidade total de ataque, c a quantidade total de pontos de velocidade. Entende-se que um personagem com poder total maior do que outro é também mais forte e mais apto no combate. Porém, esse modelo não considera a habilidade no manuseio dos controles pelo

jogador.

$$P = \sum_{n=0}^a H(n) + \sum_{n=0}^b A(n) + \sum_{n=0}^c V(n) \quad (4.3)$$

Existe uma diferença entre pontos de atributo e os valores reais que os personagens possuem. Cada ponto de atributo tem um impacto na jogabilidade e característica do personagem e eles são convertidos por uma equação linear. O sistema da Equação 4.5 mostra essa conversão, pela quantidade de pontos de atributo representado por n , no qual o atributo de vida máxima possui um incremento de 15 na vida total do personagem para cada ponto que ele tem, da mesma forma que o ataque aumenta em 5 e velocidade em 30. Um personagem sem atributo nenhum possui um valor real de 20 de vida máxima, 5 de ataque e 30 de velocidade. Esses valores foram determinados de forma arbitrária através de testes.

$$H(n) = 20 + 15 * n \quad (4.4a)$$

$$A(n) = 5 + 5 * n \quad (4.4b)$$

$$V(n) = 30 + 30 * n \quad (4.4c)$$

Assim como nos jogos incrementais, cada atributo possui um retorno linear associado. Esses valores foram determinados pelo autor através de observações com base no que foi identificado como tendo maior impacto para o resultado do combate e, como podem ser visto em Subseção 4.1.2, para a heurística de escolha de atributo na criação de um inimigo. Dessa forma, como mostrado no sistema da Equação 4.5, foi determinado que o atributo de dano possui um retorno de 1, o atributo vida máxima possui retorno de 15 e velocidade possui retorno de 150 por ponto de atributo.

$$H(n) = 15 * n \quad (4.5a)$$

$$A(n) = 1 * n \quad (4.5b)$$

$$V(n) = 150 * n \quad (4.5c)$$

Quanto maior o retorno, maior é o impacto de um ponto desse atributo para o resultado do combate. Dessa forma, foi identificado que a velocidade, seguida pela vida, possuem um impacto maior para o sucesso em combate.

Com o retorno e o custo, é possível fazer uma análise do custo-benefício desses atributos através do gráfico da Figura 4.1, que representa equações paramétricas em função das primeiras 100 melhorias, onde o eixo vertical representa o custo e o horizontal o retorno. Ambos os eixos estão em escala logarítmica de base 10.

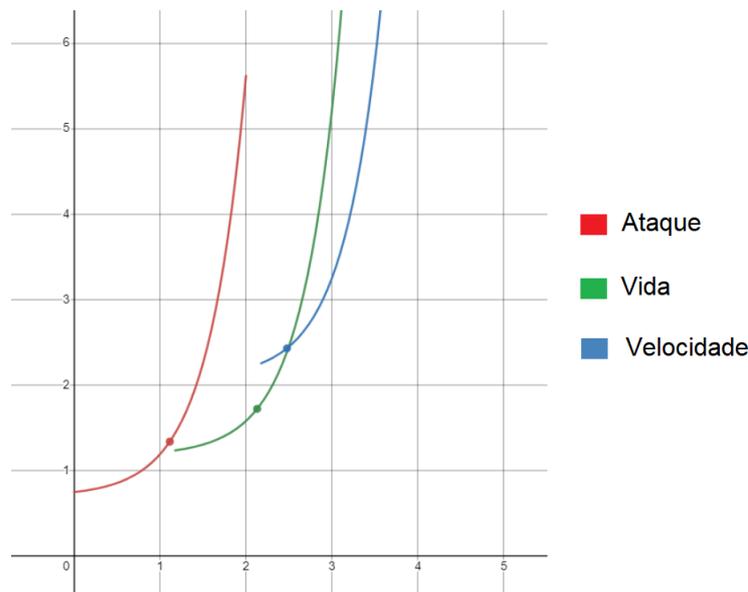


Figura 4.1: Custo benefício dos atributos. Fonte: captura de tela do PCGLab, do autor.

4.1.2 Progressão dos Inimigos

Os inimigos são criados no momento da geração do mapa, e, como explicado na Subseção 4.1.4, eles seguem uma progressão de poder que define quanto ele irá possuir. O inimigo base, o primeiro que o jogador encontra, possui um poder total de 140, com nenhum ponto de atributo. Uma vez definida a quantidade de poder, é necessário distribuí-lo nos atributos. A distribuição de poder é feita iterativamente, onde em cada iteração é escolhido um dos atributos para aumentar.

A escolha do atributo é através de uma heurística baseada no retorno estipulado de cada atributo, determinado por observações, favorecendo atributos que influenciam mais no combate. Com o valor do retorno dos atributos buscou-se controlar a

frequência com que cada atributo é escolhido. A heurística consiste em escolher o atributo cujo custo benefício, representador por CB , calculado pela Equação 4.6, for maior. Nela, o custo benefício se dá dividindo o retorno R pelo custo de melhoria do atributo C_n , onde n é a quantidade de pontos que o personagem possui daquele atributo.

$$CB = \frac{R}{C_n} \quad (4.6)$$

A Figura 4.2 exemplifica a distribuição de atributos em quatro inimigos diferentes. Todos pertencem a uma mesma sala, mas possuem poder total diferente. Além disso, a Figura 4.3 ilustra o impacto de cada atributo no somatório do poder total do inimigo. Comparando os dois gráficos, percebe-se que o impacto no poder total de um ponto do atributo velocidade é muito maior que os outros.

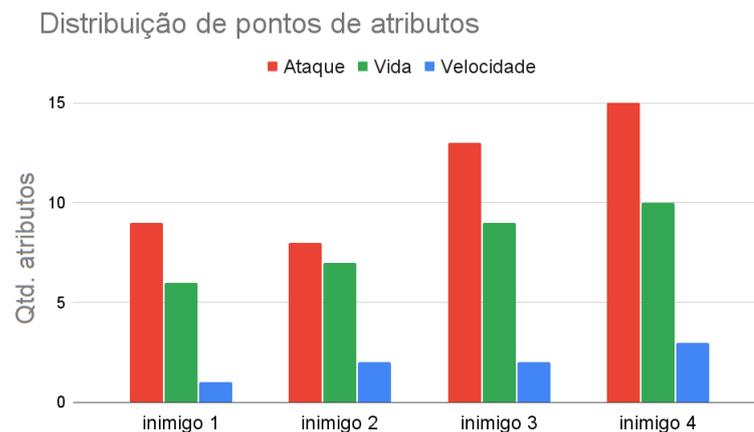


Figura 4.2: Distribuição de atributos em inimigos. Fonte: captura de tela do PCGLab, do autor.



Figura 4.3: Distribuição do poder pelos atributos de inimigos. Fonte: captura de tela do PCGLab, do autor.

4.1.3 Progressão do Jogador

A modelagem do personagem do jogador também foi feita de forma empírica através de testes pelo autor. Uma opção de *design* foi fazer com que o jogador fosse inicialmente mais rápido que os inimigos, possuindo 7 pontos do atributo velocidade, dando a ele um poder total de 5930. Um dos motivos dessa decisão foi que, por meio de observações, uma velocidade baixa tornaria a experiência do jogador difícil e entediante.

Jogadores de RPG experientes podem estranhar esse valor muito alto e distante dos inimigos iniciais. O motivo disso é que os jogos com sistema de progressão por acúmulo de pontos geralmente “escondem” esses valores altos de pontos de experiência em um nível de personagem. Depois só apresentam os pontos para o próximo nível, normalmente por uma porcentagem visível e o valor absoluto fica em segundo plano. Dessa forma, o jogador tem um controle mais imediato apenas do nível atual.

A progressão de custo de atributos do jogador é igual à dos inimigos, mas, diferente deles, é o próprio jogador que escolhe quais atributos adquirir. O único requisito é que o jogador possua pontos suficientes para adquirir o atributo desejado. Isso é feito utilizando botões na interface, como mostrado em Figura 4.4.



Figura 4.4: Interface com botões para aumentar atributos. Fonte: captura de tela do PCGLab, do autor.

Outra diferença é a maneira de se obter poder, que no caso de inimigos é atribuído um valor fixo no momento de sua criação. No caso do jogador, poder é obtido ao longo do jogo de duas formas. A primeira é através dos inimigos, que fornecem uma porcentagem de seu poder total quando o jogador os derrota. Outra maneira é ao coletar os tesouros espalhados pelo mapa.

A quantidade de poder fornecido pelos tesouros segue a mesma função de progressão de poder dos inimigos. O primeiro tesouro, o mais perto do teletransporte de entrada da primeira sala, é o tesouro base, com 100 de poder. A partir dele, segue-se uma progressão exponencial, definida pela Equação 4.7, onde n é a quantidade de tesouros posicionados no jogo até o momento, variando de 0 até a quantidade máxima de tesouros no jogo. Essa progressão determina quanto cada tesouro subsequente, em ordem de distância do teletransporte de entrada, irá fornecer de poder ao jogador. Além disso, o primeiro tesouro da próxima sala também segue a função de progressão.

$$T(n) = 100 * 1,06^{n-1} \quad (4.7)$$

4.1.4 Progressão do poder nas salas e mapa geral

Um dos objetivos buscados, seguindo a lógica de progressão, é fazer com que o jogo seja mais difícil no final do que no início, tanto no escopo geral do nível como no escopo de salas. Para isso, foi definido que quanto mais perto da saída da sala, mais forte o inimigo deve ser, e o primeiro inimigo da próxima sala também deve ser mais forte que o último inimigo da sala anterior. Dessa forma, é possível observar um crescimento contínuo de poder entre os inimigos dentro e entre salas.

A distribuição de poder é feita de sala em sala, depois que todos os inimigos foram posicionados, através do algoritmo 1. A progressão de poder dos inimigos segue a função exponencial mostrada em Equação 4.8, representado por $I(n)$, também inspirada em jogos incrementais, aplicada na linha 6 do algoritmo 1. A quantidade de poder que o n ésimo inimigo possui é determinada pela equação, onde n é a quantidade de inimigos posicionados no jogo até o momento, que varia de 0 até a quantidade de inimigos dentro do jogo.

$$I(n) = 140 * 1,06^{n-1} \quad (4.8)$$

Na Figura 4.5 é possível observar um crescimento de poder entre os inimigos de uma mesma sala, em ordem de encontro, ou seja, o primeiro é o mais perto da entrada e o último, o mais perto da saída. Essa sala, considerada grande, possui 20 inimigos,

Algoritmo 1: Distribuição de poder entre inimigos de uma sala.

```
Entrada: inimigos, numeroInimigos
Resultado: Poder distribuido para inimigos de uma sala
1 início
2   inimigosOrdenados = ordenarInimigosEmRelacaoAEntrada();
3   inimigoAtual = inimigosordenados[0];
4   for i=0 até numeroInimigos do
5     inimigoAtual = inimigosordenados[i];
6     quantidadeDePoder = determinarQuantidadePoder();
7     inimigoAtual.distribuitPoder(quantidadeDePoder);
8   end for
9 fim
```

sendo o primeiro com 505 de poder total e o último com 1623.

Já na Figura 4.6 é mostrado o crescimento de poder entre as salas. Devido à diferença da quantidade de inimigos, que varia conforme o tamanho da sala, explicado na próxima seção, o somatório de poder total dos inimigos de uma sala pode ser maior que a seguinte. Por isso, optou-se por comparar a média de poder dos inimigos entre as salas para mostrar a progressão de poder.

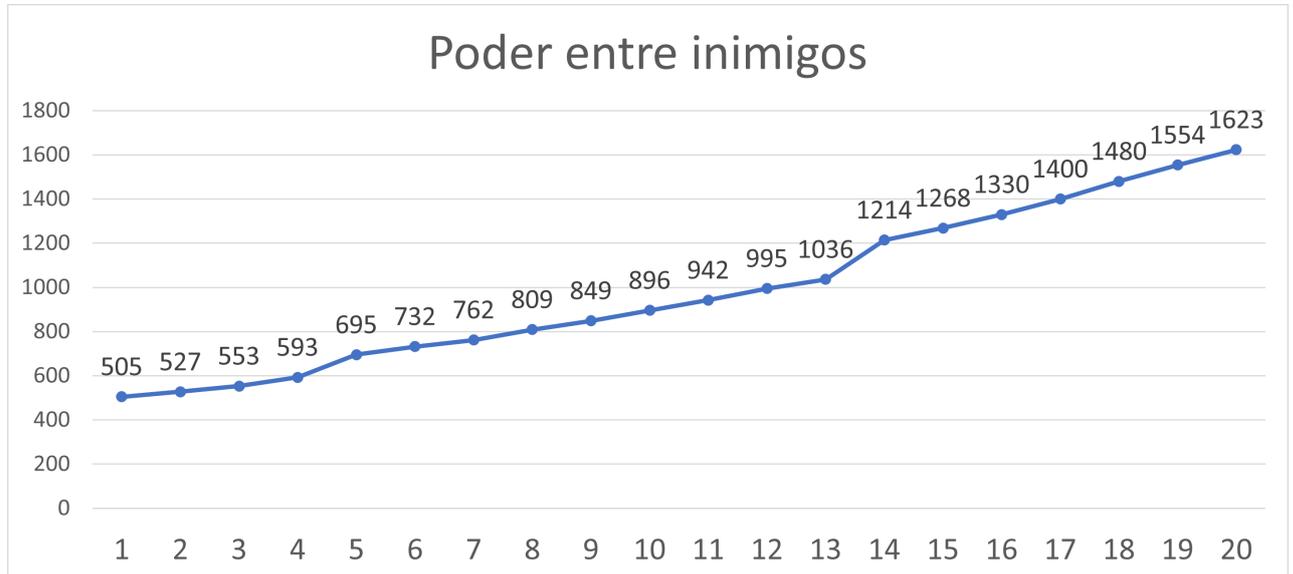


Figura 4.5: Progressão de poder entre inimigos da mesma sala. Fonte: captura de tela do PCGLab, do autor.



Figura 4.6: Progressão de poder entre salas. Fonte: captura de tela do PCGLab, do autor.

4.2 Posicionamento de elementos

Junto do sistema de progressão dos personagens, um ponto muito importante para o estudo da GPC é o posicionamento dos elementos no mapa. Em trabalhos anteriores, tanto inimigos quanto tesouros eram posicionados utilizando mapas de influência de distâncias, como explicado em Costa (2020). Porém, o atual trabalho visa unir a influência do poder de inimigos e tesouros em seus posicionamentos.

O posicionamento é feito uma sala por vez, seguindo a ordem de proximidade dos teletransportes de entrada. O algoritmo 2 mostra como é feito o posicionamento de inimigos e tesouros, que possuem características em comum. Logo, o pseudocódigo funciona para os dois elementos, e suas diferenças serão explicadas nas subseções seguintes.

A quantidade de inimigos e tesouros da sala é determinada pela Equação 4.9, em função da quantidade de células que uma sala possui. Nela, B_s é a quantidade de blocos da sala s e M é uma constante específica dos posicionamentos de inimigo e tesouro.

$$E_s = \frac{B_s}{M} \quad (4.9)$$

Enquanto a quantidade de inimigos ou tesouros de uma sala não chegar ao limite, deve-se escolher uma célula para posicioná-los. Primeiro as células são “filtradas”, descartando aquelas que não atendem aos critérios estabelecidos programados previa-

Algoritmo 2: Posicionar inimigos ou tesouros na sala.

Resultado: Elementos posicionados na sala

```

1 início
2   qtdMaxima = determinarQuantidadeMaxima();
3   qtdAtual = 0;
4   while qtdAtual < qtdMaxima do
5     celula = encontrarCelula();
6     se encontrou celula então
7       posicionar inimigo ou tesouro na celula;
8       atualizar metricas das celulas;
9       qtdAtual = qtdAtual + 1;
10    senão
11      encerrar loop
12    fim se
13  end while
14 fim

```

mente. O posicionamento de inimigo e tesouro possuem critérios específicos.

Depois, é feita uma sobreposição de alguns mapas de influência, dependendo do posicionamento. Para a combinação de métricas e mapas de influência é dado o nome de métrica composta. Esses mapas e outras métricas serão aprofundados na Seção 4.3. Para fazer a sobreposição dos mapas, são utilizados pesos individuais para fazer com que diferentes métricas tenham impacto maior ou menor no posicionamento. Essa decisão de *design* será explicada nas subseções seguintes. Com o valor do novo mapa de influência, é possível fazer uma seleção das células que são consideradas elegíveis para o posicionamento.

As células elegíveis passam por um sorteio ponderado pelo seu valor do novo mapa de influência para determinar em qual delas será posicionado o novo elemento. Quanto maior é seu valor, mais chance ela tem de ser selecionada.

4.2.1 Posicionamento de tesouros

Uma das mudanças do posicionamento de tesouros é a ordem em que ele é feito. Agora, os tesouros são posicionados no mapa antes dos inimigos. Essa foi uma decisão de projeto, no qual a ideia principal é fazer com que inimigos protejam os tesouros. Outra mudança com relação ao algoritmo anterior é a quantidade de tesouros em uma sala, que antes era a mesma da de inimigos. Se optou por diminuir a quantidade de tesouros, tornando-os

mais escasso, atribuindo à constante M da Equação 4.9 o valor de 50, ao invés de 25, como era antes.

O posicionamento de tesouros utiliza 2 mapas de influência, o primeiro deles sendo a influência de poder de tesouros (Figura 4.12), com peso de -1.5 , e a distância do caminho entrada-saída (Figura 4.11), com peso 1.5 . Os pesos foram escolhidos para fazer com que os tesouros fiquem espalhados entre si e longe do caminho principal, para motivar o jogador a explorar as salas.

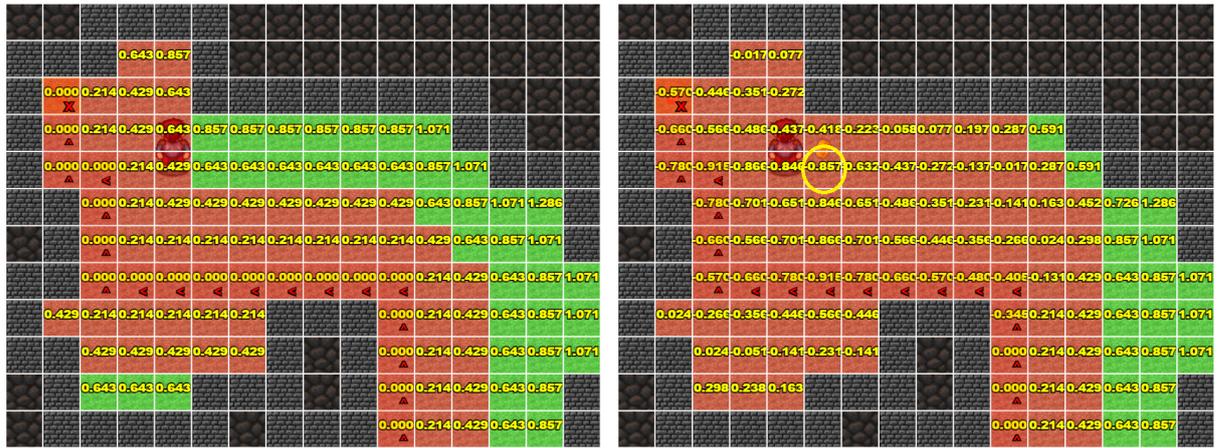
Os critérios para o posicionamento de tesouro são:

1. Estar a pelo menos 1 célula de distância de inimigos;
2. Estar a pelo menos 5 células de distância de teletransportes;
3. Estar a pelo menos 1 célula de distância de fogueiras;
4. Estar a pelo menos 4 células de distância de outros tesouros;
5. Valor da combinação do mapa de influência ser maior que 0.5.

A Figura 4.7 mostra as iterações do algoritmo de posicionamento dos tesouros. Os valores de cada célula representam o novo mapa de influência, resultado da sobreposição do mapa de poder de tesouro e do mapa de distância do caminho entrada-saída. As células em verde são as elegíveis para o posicionamento do tesouro, enquanto as vermelhas são as que não atendem aos critérios de posicionamento. Na Figura 4.7b é possível observar a influência no posicionamento quando um tesouro é posicionado na sala.

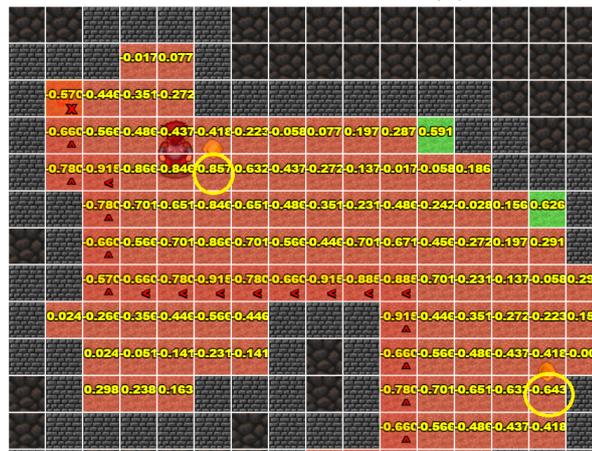
4.2.2 Posicionamento de inimigos

O *design* do posicionamento de inimigos tinha como ideia principal proporcionar desafio ao jogador, enquanto permitisse momentos de “descanso”. Para isso, foram utilizados os seguintes mapas de influência com seus respectivos pesos: distância de fogueiras (0,7), distância do caminho entrada-saída (-1), influência de poder de inimigos (-2) e influência do poder de tesouros (1,3). O objetivo era fazer com que os inimigos ficassem próximos do caminho de entrada-saída e dos tesouros, mas se afastando de teletransportes e fogueiras e entre si, para que não houvesse concentração de inimigos.



(a) Sem influência de tesouros.

(b) Um tesouro posicionado.



(c) Dois tesouros posicionados.

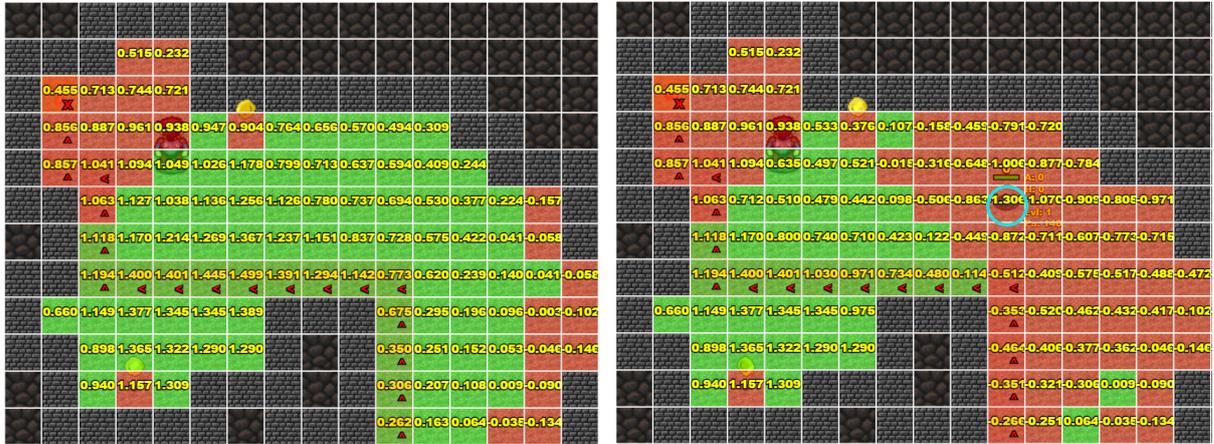
Figura 4.7: Mapas de influência de posicionamento de tesouro, onde as células verdes são elegíveis para o posicionamento e as vermelhas não atendem aos critérios. Os tesouros estão circulado em amarelo para facilitar visualização. O caminho entrada-saída está destacado pelas setas vermelhas. Fonte: captura de tela do PCGLab, do autor.

Os critérios de posicionamento de inimigos são:

1. Estar a pelo menos 5 células de distância de teletransportes;
2. Estar a pelo menos 1 célula de distância de fogueiras;
3. Estar a pelo menos 1 célula de distância de tesouros;
4. Estar a pelo menos 1 célula de distância de outros inimigos;
5. Valor da combinação do mapa de influência ser maior que 0.

A Figura 4.8 mostra as iterações do algoritmo de posicionamento dos inimigos. Os valores de cada célula representam o novo mapa de influência, resultado da sobreposição dos outros quatro já citados. As células em verde são as elegíveis para o

posicionamento inimigo, enquanto as vermelhas são as que não atendem aos critérios de posicionamento. Na Figura 4.8b é possível observar a influência no posicionamento quando um inimigo é posicionado na sala.



(a) Sem influência de inimigos.

(b) Um inimigo posicionado.



(c) Dois inimigos posicionados.

Figura 4.8: Mapas de influência de posicionamento de inimigo, onde as células verdes são elegíveis para posicionamento, enquanto vermelhas não atendem aos critérios. Fonte: captura de tela do PCGLab, do autor.

4.3 Novas métricas sobre a geração

Com o novo sistema de progressão e novas informações disponíveis, foi possível adicioná-las ao algoritmo de posicionamento dos elementos, influenciando no modelo de GPC. Além dos dados do sistema de progressão, foi adicionado também o mapa de influência da distância do caminho de entrada-saída, exemplificado na Figura 4.11.

Para o posicionamento dos elementos foi necessário combinar mapas de influência, tanto antigos como novos. Além disso, o método de combinação de mapas

utilizou soma ponderada. Diferente dos mapas de influência de distância, que aumentam em 1 para as células vizinhas, o mapa de poder possui uma propagação com decaimento, começando com o poder total do inimigo ou tesouro na célula.

Para a avaliação dos caminhos, foi adicionada uma nova métrica para medir, durante o percurso, qual o nível de desafio que o jogador encontra. Uma métrica antes utilizada para determinar se o jogador enfrentava dificuldades ao longo do caminho era a distância de inimigos, mas com o sistema de progressão de dificuldade, é interessante analisar se esse desafio aumenta ao longo do tempo. O mapa de influência de poder dos inimigos fornece a informação necessária, como mostrado na Figura 4.9, ilustrado por meio da linha azul do gráfico da Figura 4.10. Os gráficos com as métricas dos caminhos possuem o eixo vertical normalizado, e em suas legendas é mostrado o valor máximo atingido por cada métrica.



Figura 4.9: Mapa de influência de poder dos inimigos. Fonte: captura de tela do PCGLab, do autor.

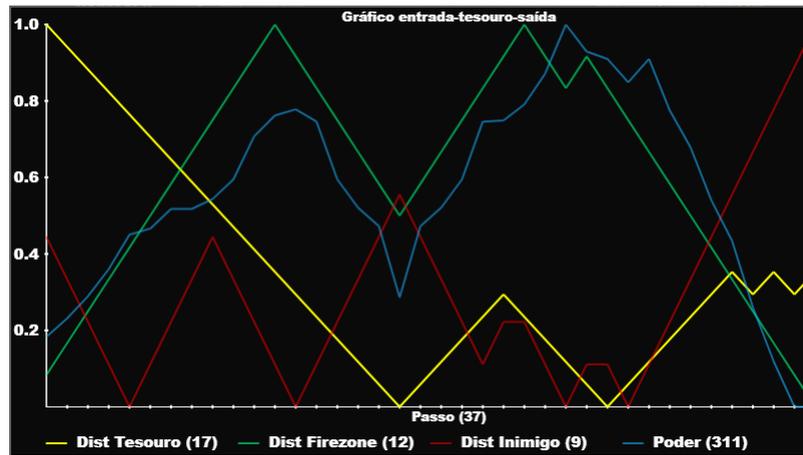


Figura 4.10: Gráfico com métricas de um caminho entrada-tesouro-saída. Fonte: captura de tela do PCGLab, do autor.

4.4 Avaliação da geração

Para análise do sistema de progressão e do novo posicionamento dos elementos, será realizada uma comparação do posicionamento antigo e do novo, através da avaliação da geração da sala e seus caminhos de entrada-saída e o de entrada-tesouro-saída, que passa por todos os tesouros, com seus respectivos gráficos. Os gráficos mostram a distância dos elementos para cada passo do caminho, gerados em tempo de execução do jogo.

A avaliação foi separada por tamanho de salas. Foram consideradas salas pequena, média e grande. A análise será feita evidenciando-se as diferenças encontradas nas gerações que impactam na experiência do jogador, e não por meio de avaliações individuais de cada caminho.

4.4.1 Sala Pequena

A Figura 4.13 mostra o caminho entrada-saída e seus respectivos gráficos, tanto do posicionamento antigo quanto do novo, colocados lado a lado. Da mesma forma, a Figura 4.14 ilustra o caminho de entrada-tesouro-saída.

Devido ao pouco espaço da sala, a mudança do posicionamento em salas pequenas pode não resultar em diferenças significativas na experiência do jogador. Ainda assim, na sala em questão, é possível observar no posicionamento antigo que um dos inimigos, destacado na Figura 4.13a, se encontra longe o suficiente do caminho para que o

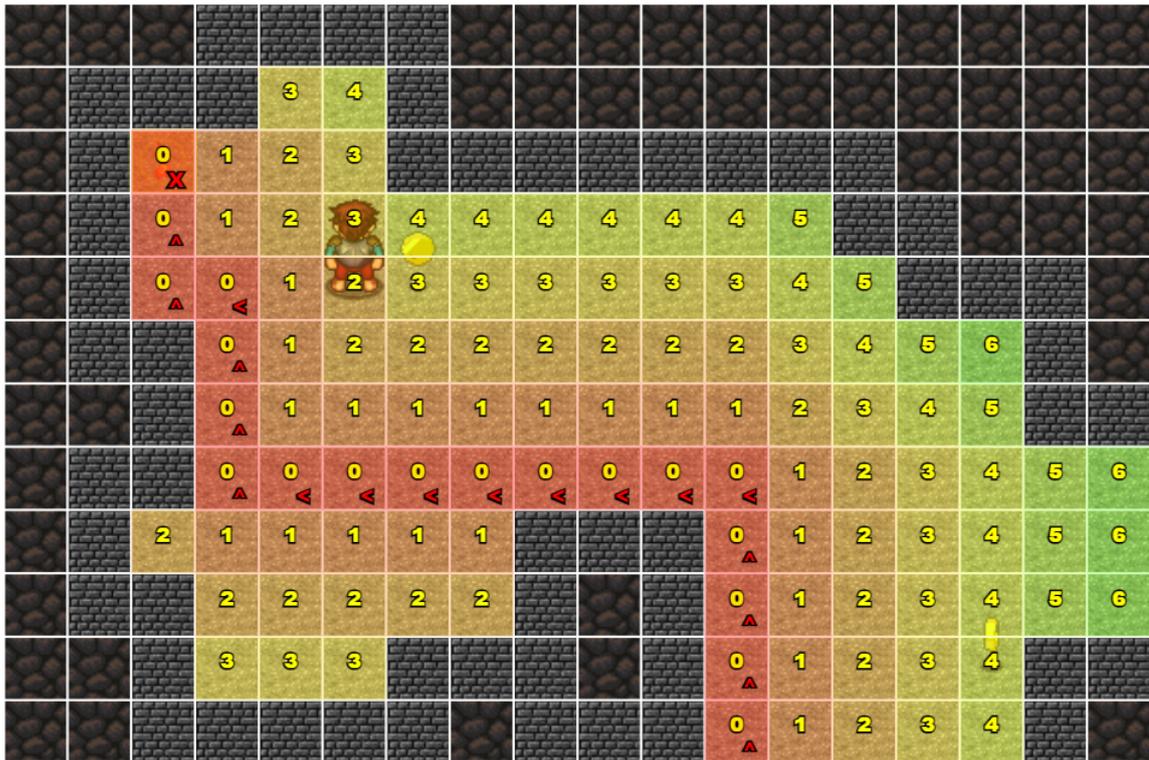


Figura 4.11: Mapa de influência do caminho entrada-saída. Fonte: captura de tela do PCGLab, do autor.

jogador consiga passar por ele sem enfrentá-lo. Já no novo posicionamento, ilustrado na Figura 4.13b, o jogador terá que enfrentar os dois inimigos para atravessar a sala. Dessa forma, existe a diferença do número de inimigos que o jogador pode enfrentar caso escolha por percorrer o caminho mínimo.

Em relação ao posicionamento do tesouro, é possível notar que no posicionamento antigo ele é o primeiro elemento que o jogador encontra. Já no posicionamento novo, só é possível coletá-lo após passar pelos inimigos. Pelo gráfico, ilustrado na Figura 4.14d, observa-se que antes e próximo ao passo que entra em contato com o tesouro, o jogador passa por uma área de pico de poder, enquanto no posicionamento antigo, na Figura 4.14c, esse pico de poder acontece apenas depois. Mesmo que para passar para a próxima sala seja necessário enfrentar os inimigos em seguida, a disposição dos elementos têm diferença devido ao poder que o tesouro fornece ao jogador. No posicionamento antigo, o jogador recebe o poder, ou seja, a recompensa, sem antes passar pelo desafio.

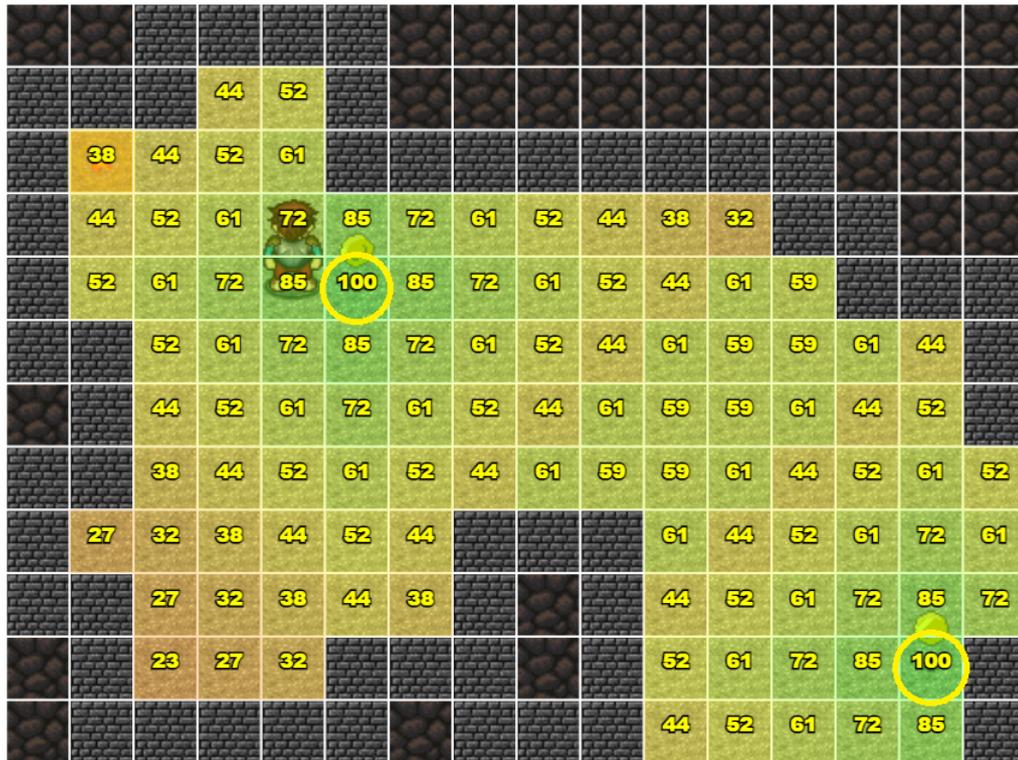


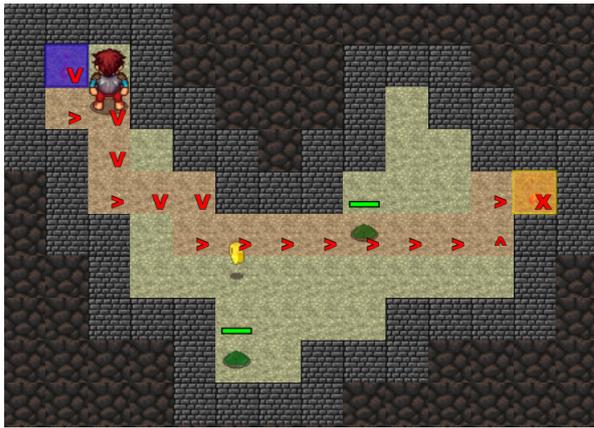
Figura 4.12: Mapa de influência do poder de tesouros Fonte: captura de tela do PCGLab, do autor.

4.4.2 Sala Média

Seguindo o mesmo tipo de comparação, a Figura 4.15 mostra o caminho entrada-saída e seus respectivos gráficos, tanto do posicionamento antigo, à esquerda, quanto do novo, à direita. Da mesma forma, a Figura 4.16 ilustra o caminho de entrada-tesouro-saída.

Com relação ao posicionamento dos inimigos, é possível notar que em ambos algoritmos existe um inimigo no canto superior direito da sala que não oferece ameaça para o jogador, como mostrado nas figuras do mapa da Figura 4.15a e da Figura 4.15b, uma vez que não é necessário passar perto dele para coletar tesouros ou avançar para a próxima sala. Além disso, observa-se que, na Figura 4.16b, os inimigos se concentraram perto de um dos tesouros, enquanto o outro ficou "desprotegido", numa área de baixa influência de poder. Apesar do novo algoritmo tentar resolver esses cenários, ainda é possível que estes ocorram devido ao fator de aleatoriedade.

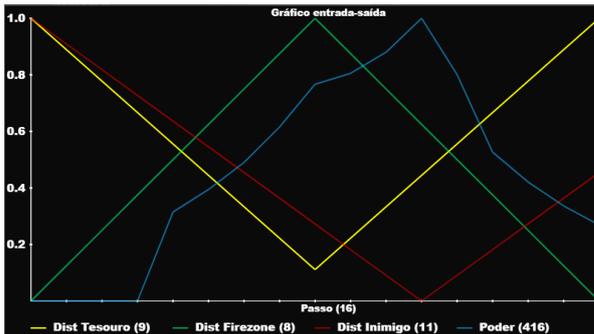
Em relação aos tesouros, vale ressaltar que o gráfico da Figura 4.16c possui 3 passos em que se passa por um tesouro, mas isso ocorre porque o caminho passa duas vezes na célula de um dos tesouros. Analisando seus posicionamentos, é possível notar que



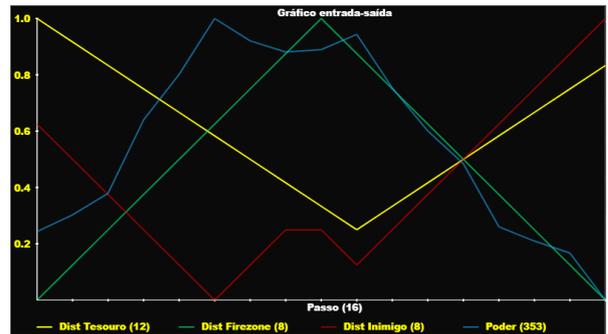
(a) Caminho do posicionamento antigo.



(b) Caminho do novo posicionamento.



(c) Gráfico do posicionamento antigo.



(d) Gráfico do novo posicionamento.

Figura 4.13: Caminho da entrada até a saída do algoritmo de posicionamento antigo e novo, junto de seus gráficos, da sala pequena. Fonte: captura de tela do PCGLab, do autor.

eles não mais foram posicionados no caminho mínimo de entrada-saída, como mostrado nos gráficos de caminhos, na Figura 4.15c e na Figura 4.15d. No posicionamento antigo, é possível coletar um tesouro seguindo o caminho mínimo, já no novo algoritmo nenhum tesouro é coletado seguindo o mesmo caminho.

Ainda em relação aos tesouros, assim como ocorreu na sala pequena, nota-se que não mais o jogador consegue coletá-los sem antes superar os desafios, como é possível observar comparando o posicionamento na Figura 4.16a e na Figura 4.16b. Analisando os gráficos do caminho entrada-tesouro-saída, na Figura 4.16c e na Figura 4.16d, percebe-se que o novo algoritmo proporciona muito mais encontros diretos com os inimigos. Apesar disso, para o mapa em questão, o caminho do posicionamento antigo ainda levaria o jogador a engajar em combate com os inimigos, por conta da proximidade de dois blocos.

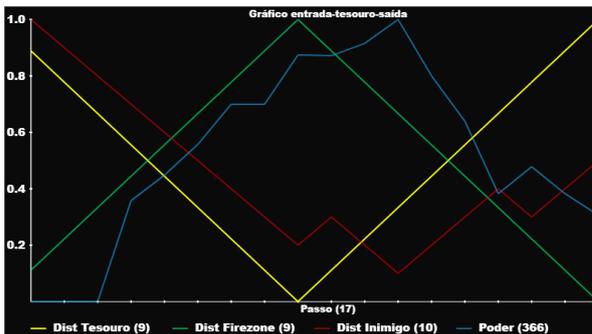
Além disso, a disposição dos inimigos em relação ao caminho entrada-tesouro-saída do novo posicionamento possui um pico de poder maior, de 310, comparado com os 224 do posicionamento antigo. Apesar do jogador ter que enfrentar a mesma quantidade



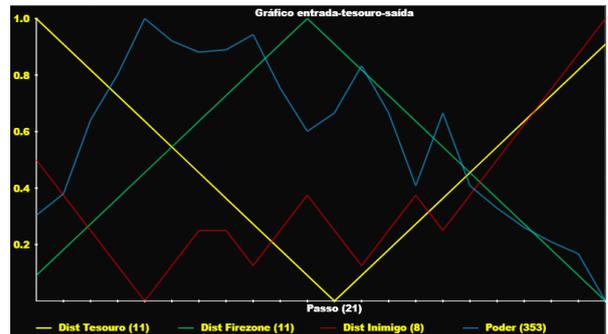
(a) Caminho do posicionamento antigo.



(b) Caminho do novo posicionamento.



(c) Gráfico do posicionamento antigo.



(d) Gráfico do novo posicionamento.

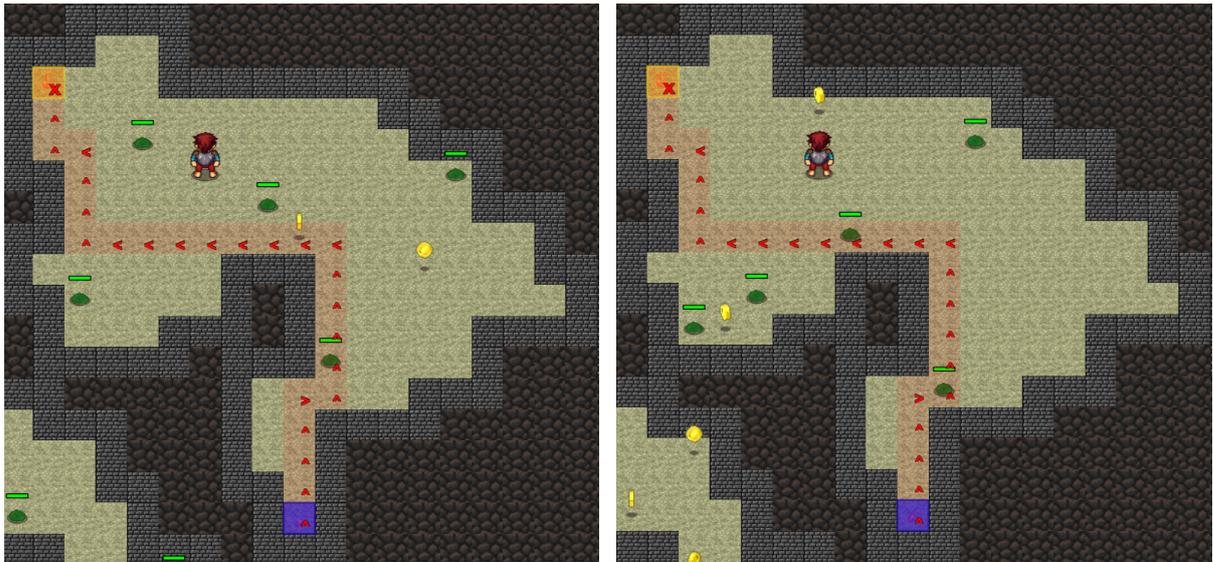
Figura 4.14: Caminho que passa por todos os tesouros do algoritmo de posicionamento antigo e novo, junto de seus gráficos, da sala pequena. Fonte: captura de tela do PCGLab, do autor.

de inimigos para coletar os tesouros em ambos mapas gerados, a concentração de inimigos no novo posicionamento aumenta a influência de poder. Com isso é possível concluir que o jogador precisa enfrentar um desafio maior para coletar todas as recompensas no novo algoritmo.

4.4.3 Sala Grande

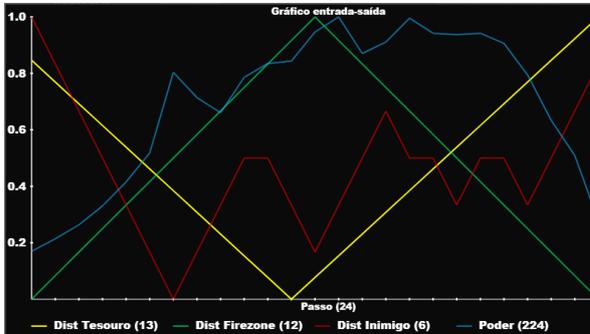
A Figura 4.17 mostra o caminho entrada-saída e seus respectivos gráficos, tanto do posicionamento antigo, à esquerda, quanto do novo, à direita. Da mesma forma, a Figura 4.18 ilustra o caminho de entrada-tesouro-saída.

Com relação ao posicionamento dos inimigos, comparando-se os mapas da Figura 4.17a e da Figura 4.17b, observa-se que existe uma área de “descanso” logo no início da sala gerada pelo novo algoritmo de posicionamento, que não está presente no mapa do posicionamento antigo. Ainda em relação aos inimigos, observa-se através dos

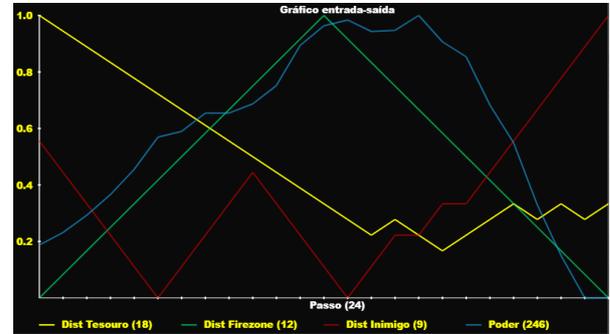


(a) Caminho do posicionamento antigo.

(b) Caminho do novo posicionamento.



(c) Gráfico do posicionamento antigo.



(d) Gráfico do posicionamento novo.

Figura 4.15: Caminho da entrada até a saída dos algoritmos de posicionamento antigo e novo, junto de seus gráficos, da sala média. Fonte: captura de tela do PCGLab, do autor.

caminhos entrada-tesouro-saída, da Figura 4.18a e da Figura 4.18b, e por seus gráficos, na Figura 4.18c e na Figura 4.18d, que eles foram posicionados de maneira a proteger os tesouros e não mais espalhados em locais que não oferecem ameaça ao jogador.

Além disso, novamente é possível perceber, comparando-se os gráficos do caminho de entrada-saída, da Figura 4.17c e da Figura 4.17d, que o jogador não encontra tesouros no caminho mínimo. Para que ele consiga coletar tesouros e acumular mais poder, é necessário que ele explore a sala e enfrente inimigos que, caso contrário, não seria necessário enfrentar.

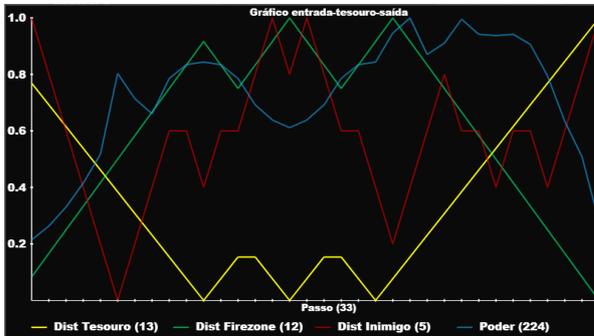
No tocante ao desafio presente na sala, apesar de no gráfico do caminho entrada-tesouro-saída do novo algoritmo, representado na Figura 4.18d apresentar como pico de poder um valor de 3739, menor quando comparado aos 4047 do posicionamento antigo (Figura 4.18c), é possível observar que o novo posicionamento possui vários momentos



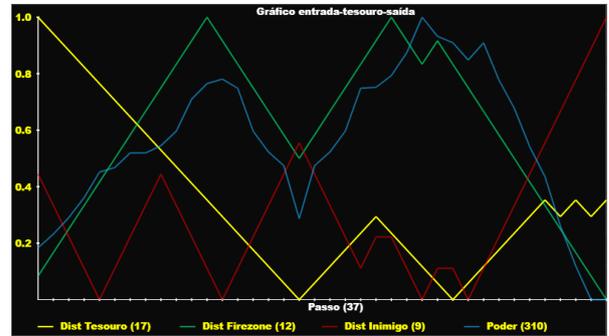
(a) Caminho do posicionamento antigo.



(b) Caminho do novo posicionamento.



(c) Gráfico do posicionamento antigo.



(d) Gráfico do posicionamento novo.

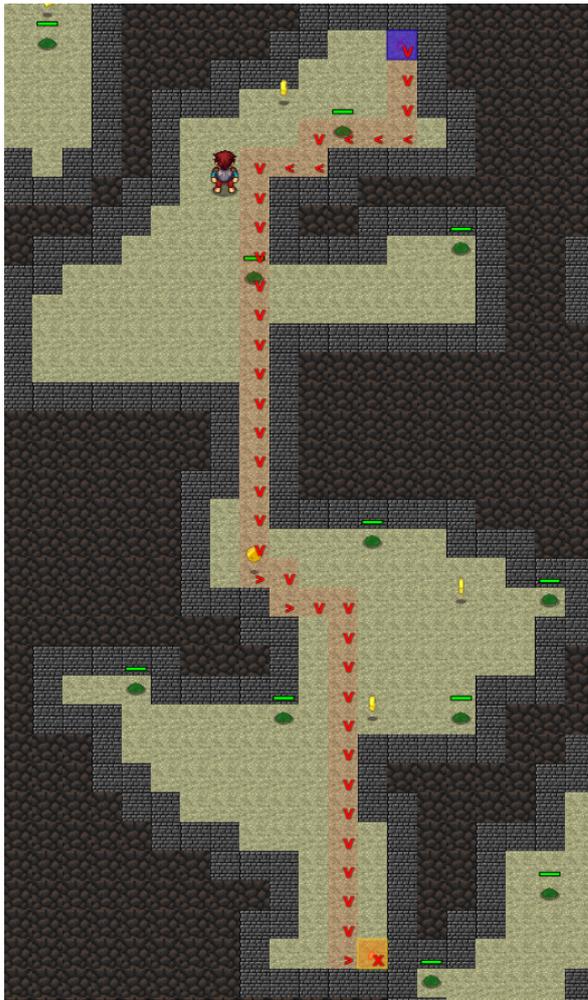
Figura 4.16: Caminho que passa por todos os tesouros do algoritmo de posicionamento antigo e novo, junto de seus gráficos, da sala média. Fonte: captura de tela do PCGLab, do autor.

de tensão para que o jogador consiga coletar todas as recompensas. No posicionamento antigo, duas vezes o tesouro é coletado em áreas de baixa influência de poder.

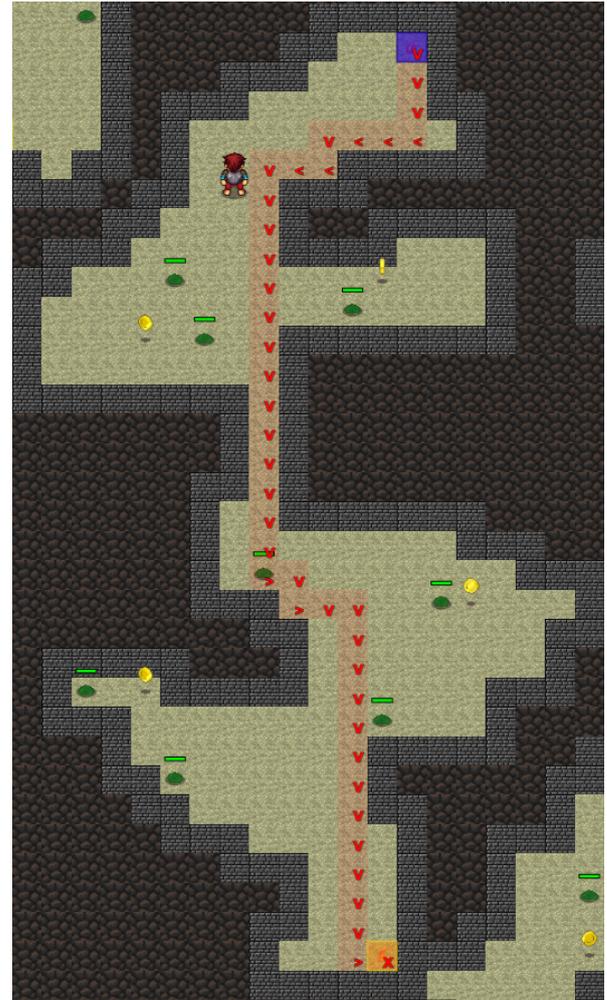
4.4.4 Conclusão da avaliação

Com relação ao posicionamento dos elementos, é possível perceber pelos mapas gerados, de todos os tamanhos de sala, que os tesouros não mais se encontram no caminho mínimo de entrada-saída. Esse efeito é positivo, pois diminui a ocorrência de regiões sem conteúdo, uma vez que agora há maior chance dos tesouros serem espalhados em regiões mais afastadas da sala, além de fazer com que o jogador tenha um incentivo para explorar a sala.

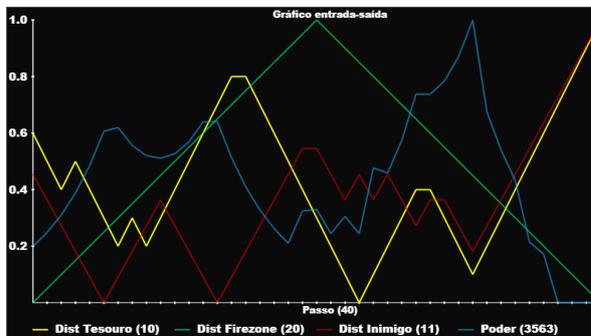
Com relação aos inimigos, um dos objetivos era fazer com que eles oferecessem



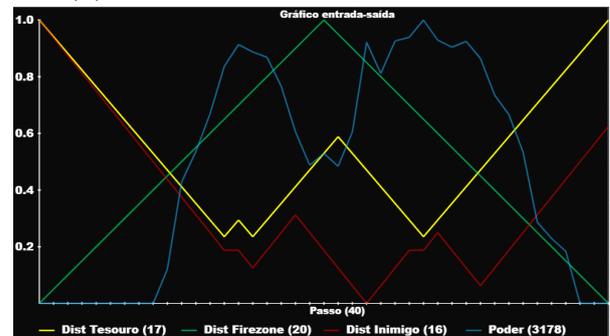
(a) Caminho do posicionamento antigo.



(b) Caminho do novo posicionamento.



(c) Gráfico do posicionamento antigo.



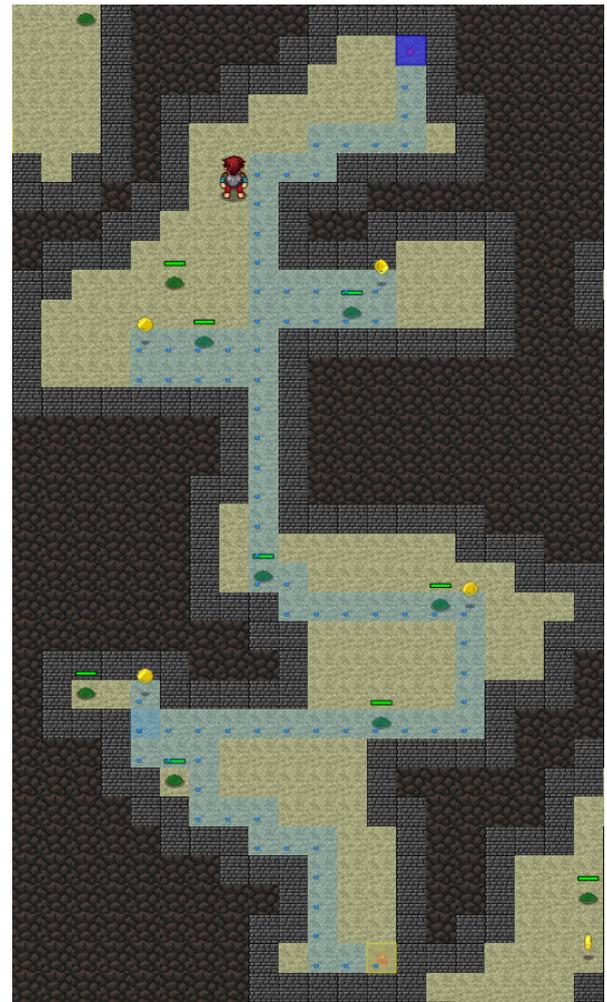
(d) Gráfico do novo posicionamento.

Figura 4.17: Caminho da entrada até a saída dos algoritmos de posicionamento antigo e novo, junto de seus gráficos, da sala grande. Fonte: captura de tela do PCGLab, do autor.

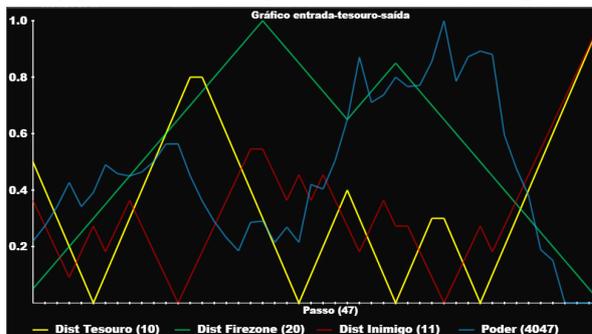
desafio ao jogador antes dele conseguir uma recompensa, como coletar um tesouro ou chegar na próxima sala, ao ficar no caminho entrada-saída e em volta de tesouros, enquanto se evitasse a concentração dos mesmos. É possível observar, através dos gráficos do caminho que passa pelos tesouros, que existe uma maior influência de poder ao redor dos tesouros, principalmente antes de coletá-lo. Apesar disso, como visto na sala média,



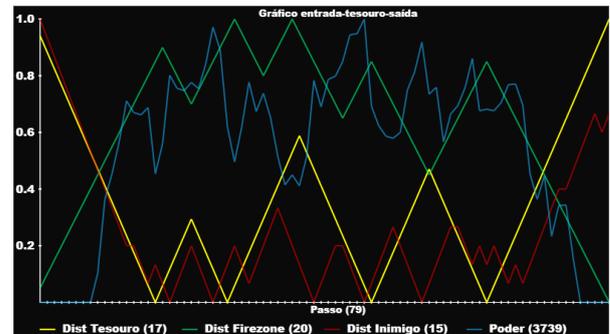
(a) Caminho do posicionamento antigo.



(b) Caminho do novo posicionamento.



(c) Gráfico do posicionamento antigo.



(d) Gráfico do novo posicionamento.

Figura 4.18: Caminho que passa por todos os tesouros dos algoritmos de posicionamento antigo e novo, junto de seus gráficos, da sala grande. Fonte: captura de tela do PCGLab, do autor.

ainda poder ocorrer de um inimigo ser posicionado em um lugar remoto, sem oferecer ameaça ao jogador.

No que tange o desafio, é possível observar através dos gráficos, que a influência de poder possui muito mais picos, principalmente no caminho entrada-tesouro-saída de salas maiores. Esse era um dos efeitos desejados para proporcionar uma maior dificuldade

para o jogador, que muitas vezes conseguia passar por uma sala sem quase entrar em combate.

Este capítulo apresentou o desenvolvimento do modelo de progressão e sua utilização no posicionamento dos elementos durante a geração do mapa e análise dos seus possíveis impactos na experiência do jogador, assim como a criação de novas métricas para avaliação.

5 Considerações Finais

Este trabalho apresentou a criação de um modelo de progressão aplicado à Geração Procedural de Conteúdo (GPC) e analisou seu impacto na experiência do jogador. O modelo teve como base jogos incrementais, onde foi realizada uma adaptação de suas melhorias para os atributos dos personagens do sistema desenvolvido.

Em relação aos objetivos específicos do trabalho, foi possível criar um modelo de progressão de dificuldade, onde tanto os inimigos quanto o jogador fazem uso desse modelo, e também foi dada ao jogador a escolha de como seu personagem aumenta seu poder, através da escolha de atributos. Além disso, a geração e o posicionamento espacial dos elementos de jogo utilizaram esse modelo, sendo possível analisar os efeitos na experiência do jogador.

Um dos objetivos específicos do trabalho, de analisar a progressão e o posicionamento dos elementos com um jogador real, para avaliar sua experiência, tanto por meio de métricas quanto por formulários de aceitação, não foi alcançada. Apesar disso, a ferramenta dispõe de diversas métricas que permitem, em um trabalho futuro, observar e analisar o comportamento do jogador, e realizar ajustes nos parâmetros dos sistemas para tentar colocar o jogador no estado de *flow*.

Uma das limitações encontradas por trabalhos passados era a falta de dificuldade enfrentada pelo jogador, que podia influenciar no caminho que ele percorre. Agora, ele precisa ser muito mais cuidadoso ao avançar, uma vez que não é mais possível passar por inimigos fortes sem sofrer dano, exceto se antes acumular poder.

Outro problema apontado anteriormente era a existência de recompensas que o jogador poderia coletar sem enfrentar nenhum desafio. Com o posicionamento atual, a influência de poder tende a colocar os inimigos próximos às recompensas, ao passo que as recompensas são posicionadas longe do caminho mínimo entrada-saída, apesar de em alguns momentos isso não ocorrer.

Com o desenvolvimento do sistema de progressão em torno da variável poder, foi possível adicionar ao posicionamento dos elementos a influência dessa variável, tanto

de inimigos quanto de tesouro, além de construir cenários onde o jogador é incentivado a explorar as salas para acumular poder.

Além disso, a criação das novas métricas, em conjunto com as já existentes anteriormente, constituiu um ferramental no ambiente do PCGLab para avaliar a experiência do jogador e seu progresso, assim como maneiras de ajustar o jogo para deixá-lo mais fácil ou difícil.

5.1 Limitações e Trabalhos Futuros

Enquanto acreditamos que o trabalho se aproximou dos objetivos geral e específicos, algumas limitações não permitiram uma melhor avaliação da progressão do jogador. Dentre elas, os mapas de influência serem “estáticos” e não terem seus valores atualizados ao longo do jogo. Os mapas são criados durante a criação do nível, e mesmo quando os inimigos se movem, por exemplo, as influências dos mesmos continuam como antes. Apesar de não influenciar no posicionamento dos elementos, poderia ajudar na análise do caminho percorrido pelo jogador.

Outra limitação do trabalho é o combate muito simples e a falta de diversidade de inimigos. Foi necessário simplificar a modelagem dos personagens para o sistema de progressão, incluindo apenas 3 atributos dos mesmos, sendo que eles possuem mais. Além disso, devido à falta de variedade de mecânicas e de diferentes tipos de inimigos, não foi possível analisar a variação do nível de desafio de habilidade, que permanece o mesmo durante todo o jogo.

Como trabalho futuro, seria interessante tornar o combate mais complexo, com diferentes tipos de inimigos e adição de novos atributos para compor o sistema de progressão. Dessa forma, seria possível criar diferentes classes de encontro, desafiando o jogador com novas mecânicas e enriquecendo sua experiência.

Outra proposta de trabalho futuro é a criação de diferentes níveis de dificuldade. Um dos problemas em progressão em jogos apontados por Schreiber e Romero (2021) é que cada jogador possui uma habilidade prévia diferente. Com o desenvolvimento desses níveis, é possível fazer com que um jogador mais habilidoso tenha uma experiência tão imersiva quanto um jogador com pouca habilidade. Caso contrário, corre o risco deles

não entrarem no estado de *flow*. Outra abordagem para o mesmo problema é fazer com que a dificuldade seja adaptativa, de modo que o jogo, em tempo de execução, analise o desempenho do jogador e ajuste a dificuldade.

Em relação à geração dos mapas, foi testado o impacto da topologia gerada na progressão do jogador. É interessante também aplicar o conhecimento adquirido na análise da progressão e experiência do jogador na geração do mapa e no posicionamento dos elementos. Além disso, é possível adicionar novas métricas para avaliação da GPC, como, por exemplo, a distância que um bloco tem da parede da sala, para identificar tipos de regiões (sala grande, corredor estreito) e criar diferentes tipos de encontros com inimigos.

Bibliografia

- ADAMS, E. *Fundamentals of game design*. [S.l.]: Pearson Education, 2014.
- APPERLEY, T. Genre and game studies: Toward a critical approach to video game genres. *Simulation & Gaming*, v. 37, p. 23 – 6, 2006.
- BATCHELOR, J. *GamesIndustry.biz presents... The Year in Numbers 2020*. gamesindustry, 2020. Disponível em: <https://www.gamesindustry.biz/articles/2020-12-21-gamesindustry-biz-presents-the-year-in-numbers-2020>).
- BRATHWAITE, B.; SCHREIBER, I. *Challenges for Game Designers*. 1. ed. USA: Charles River Media, Inc., 2008. ISBN 158450580X.
- BREWER, N. *GOING ROGUE: A BRIEF HISTORY OF THE COMPUTERIZED DUNGEON CRAWL*. 2016. Disponível em: <https://insight.ieeeusa.org/articles/going-rogue-a-brief-history-of-the-computerized-dungeon-crawl/>).
- BYCER, J. *Understanding Progression Models in Game Design*. 2018. Disponível em: <https://game-wisdom.com/critical/progression-models>).
- CHIOVATO, L. *Game Live Streaming: What Is Live Streaming? How Big Is the Audience? How Did the Pandemic Impact Live Streaming?* NewZoo, 2021. Disponível em: <https://newzoo.com/insights/articles/what-is-game-live-streaming-how-big-is-the-audience-pandemic-impact-twitch-youtube/>).
- COSTA, L. da. *Geração procedural de conteúdo através de uma abordagem híbrida entre autômatos celulares e heurísticas*. 2020. Disponível em: <https://www.sbgames.org/proceedings2020/WorkshopG2/209761.pdf>).
- CSIKSZENTMIHALYI, M. et al. *Flow*. [S.l.]: New York: Harper & Row, 1990.
- DORMANS, J. *Engineering emergence: applied theory for game design*. [S.l.]: University of Amsterdam, 2013.
- DUMATIVA; BROTHERS, C. *A Lenda do Herói*. 2016. Disponível em: <http://alendadoheroi.com.br/>).
- KING, A. *Numbers Getting Bigger: The Design and Math of Incremental Games*. 2015. Disponível em: <https://gamedevelopment.tutsplus.com/articles/numbers-getting-bigger-the-design-and-math-of-incremental-games--cms-24023>).
- KING, A. *Numbers Getting Bigger: What Are Incremental Games, and Why Are They Fun?* 2015. Disponível em: <https://gamedevelopment.tutsplus.com/articles/numbers-getting-bigger-what-are-incremental-games-and-why-are-they-fun--cms-23925>).
- KUITTINEN, P. *Rogue - Exploring the Dungeons of Doom (1980)*. 2001. Disponível em: <https://web.archive.org/web/20071217100401/http://users.tkk.fi/~eye/roguelike/rogue.html>).

- MARK, D. Modular tactical influence maps. In: *Game AI Pro 360*. [S.l.]: CRC Press, 2019. p. 103–124.
- MOBYGAMES. *Glossário*. 2021. Disponível em: <https://www.mobygames.com/glossary/genres/>.
- MOSS, R. C. *ASCII art + permadeath: The history of roguelike games*. 2020. Disponível em: <https://arstechnica.com/gaming/2020/03/ascii-art-permadeath-the-history-of-rogue-like-games/>.
- OSBORN, J. C. et al. *Combat in games*. 2018.
- ROUSSEAU, J. *Brazilian games market to hit USD2.3bn in 2021 - Newzoo*. gamesindustry, 2021. Disponível em: <https://www.gamesindustry.biz/brazilian-games-market-to-hit-23bn-in-2021-newzoo>.
- SANTANA, G. M. *Avaliação do Fluxo da Experiência do Jogador na Geração Procedural por Autômatos Celulares*. 2021. Available at <http://monografias.ice.ufjf.br/tcc-web/tcc?id=559>.
- SCHREIBER, I.; ROMERO, B. *Game balance*. [S.l.]: CRC Press, 2021.
- SILVA, L. P. *Rogue (PC) e o nascimento de um gênero*. 2015. Disponível em: <https://www.gameblast.com.br/2015/05/rogue-pc-e-o-nascimento-de-um-genero.html>.
- TAKAHASHI, D. *SuperData: Games grew 12% to USD139.9 billion in 2020 amid pandemic*. venturebeat, 2021. Disponível em: <https://venturebeat.com/2021/01/06/superdata-games-grew-12-to-139-9-billion-in-2020-amid-pandemic>.
- TOGELIUS, J. et al. What is procedural content generation?: Mario on the borderline. In: *ACM. Proceedings of the 2nd international workshop on procedural content generation in games*. [S.l.], 2011. p. 5.
- TOGELIUS, J. et al. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, Institute of Electrical and Electronics Engineers, Inc.,— y USA United States, v. 3, n. 3, p. 172–186, 2011.
- TOZOUR, P. *Game Programming Gems 2*. [S.l.]: Charles River Media, 2001.
- VERDE, R. C. V. *Avaliação da geração de conteúdo por Wave Function Collapse na experiência do jogador*. 2021. Available at <http://monografias.ice.ufjf.br/tcc-web/tcc?id=572>.
- VIRGO, D. *2020 Essential Facts about Video Game Industry*. digitalvirgo, 2020. Disponível em: <https://www.digitalvirgo.com/video-game-industry/>.
- WIJMAN, T. *Global Games Market to Generate \$175.8 Billion in 2021; Despite a Slight Decline, the Market Is on Track to Surpass \$200 Billion in 2023*. NewZoo, 2021. Disponível em: <https://newzoo.com/insights/articles/global-games-market-to-generate-175-8-billion-in-2021-despite-a-slight-decline-the-market-is-on-track-to-surpass-200-billion-in-2023/>.

WIJMAN, T. *The Games Market Will Decline -4.3% to \$184.4 Billion in 2022; Long-Term Outlook Remains Positive*. NewZoo, 2022. Disponível em: <https://newzoo.com/insights/articles/the-games-market-will-decline-4-3-to-184-4-billion-in-2022#:~:text=We%20forecast%20the%20console%20games,AAA%20releases%20hurts%20the%20market.>

WINKIE, L. *People who argue about the definition of roguelikes are annoying, but what if they're right?* 2021. Disponível em: <https://www.pcgamer.com/people-who-argue-about-the-definition-of-roguelikes-are-annoying-but-what-if-theyre-right/>.

WOLF, M. *The Video Game Explosion: A History from PONG to Playstation and Beyond*. Greenwood Press, 2008. 259 p. ISBN 9780313338687. Disponível em: <https://books.google.com.br/books?id=XiM0ntMybNwC>.

ZAPATA, S. *On the Historical Origin of the “Roguelike” Term*. 2017. Disponível em: <https://blog.slashie.net/on-the-historical-origin-of-the-roguelike-term/>.