

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Ambiente para Planejamento e Simulação de Dificuldade em Jogos

João Cotta Badaró

JUIZ DE FORA
JULHO, 2023

Ambiente para Planejamento e Simulação de Dificuldade em Jogos

JOÃO COTTA BADARÓ

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da computação

Orientador: Igor de Oliveira Knop

JUIZ DE FORA
JULHO, 2023

AMBIENTE PARA PLANEJAMENTO E SIMULAÇÃO DE DIFICULDADE EM JOGOS

João Cotta Badaró

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 CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Igor de Oliveira Knop
D.Sc. Modelagem Computacional

Marcelo Caniato Renhe
Doutor em Engenharia de Sistemas e Computação

Pedro Henrique Dias Valle
Doutor em Ciências da Computação e Matemática Computacional

JUIZ DE FORA
14 DE JULHO, 2023

Aos meus amigos, irmãos e a minha namorada.

Aos pais, pelo apoio e sustento.

Resumo

A dificuldade em jogos é um conceito associado à capacidade do jogador superar os desafios impostos, seja por habilidade nos controles ou por acúmulo de poder dentro do jogo, e está diretamente ligada à experiência vivenciada durante uma partida. Pessoas possuem diferentes habilidades, cognitivas ou motoras, e os obstáculos encontrados podem não ser agradáveis a todos. Dessa forma, surge a necessidade de um planejamento e controle de dificuldade por parte do projetista para conseguir manter os jogadores engajados durante todo o progresso do conteúdo. Estimular o jogador a entrar na zona do *flow*, equilibrando o desafio com a perícia observada na atividade tem sido um dos principais objetivos em cada projeto da indústria de jogos. Este trabalho apresenta uma pesquisa exploratória através do projeto e avaliação de um simulador para modelos de dificuldade. A ferramenta proposta possibilita representar a navegabilidade no espaço de desafios do jogo, perfis de jogador, modelos de desistência e progressão de habilidade. Espera-se que pelo uso de métricas sobre os dados simulados, seja possível comparar o efeito de diferentes heurísticas e modelos em busca de um controlador de dificuldade adaptativa, aumentando o conhecimento sobre o assunto e dando base para outras pesquisas sobre a experiência de jogador em diversos gêneros de jogos.

Palavras-chave: desenvolvimento de jogos; game design; dificuldade; flow.

Abstract

Difficulty in games is a concept associated with the player's ability to overcome the challenges imposed, either by skill in controlling the game or by accumulating power within it. This concept is directly linked to the experience lived during a match. As people have different cognitive or motor abilities, the obstacles encountered may not be enjoyable for everyone. Therefore, game designers need to plan how difficulty is implemented in games to engage players throughout their progress. Encouraging the player to enter the *flow* zone, where the challenge is balanced with their expertise, has been one of the main objectives in every game development project. This work presents an exploratory research project through the design and evaluation of a simulator for difficulty models. The proposed tool enables the representation of navigability in the space of game challenges, player models, quitting models, and skill progression. By using metrics on simulated data, it is expected that the effect of different heuristics and models can be compared in the search for an adaptive difficulty controller. This will increase knowledge on the subject and provide a basis for further research on the player experience in different game genres.

Keywords: game development; game design; difficulty; flow.

Agradecimentos

A todos os meus amigos e família e a minha namorada Geisa, eu nunca teria conseguido terminar isso sem vocês, muito obrigado. Especialmente a Pedro, Victor e Daniel o apoio de vocês foi incrível, obrigado por esses anos de aprendizagem e felicidade.

Ao professor Igor Knop pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

“A nossa vida é aquilo que os nossos pensamentos fizeram dela.”

Marco Aurélio

Conteúdo

Lista de Figuras	8
1 Introdução	11
1.1 Apresentação do Tema	11
1.2 Contextualização	11
1.3 Descrição do Problema	12
1.4 Justificativa e Motivação	13
1.5 Questões de Pesquisa e Hipótese	14
1.6 Objetivos	14
1.7 Metodologia	15
1.8 Organização do texto	16
2 Fundamentação Teórica	17
2.1 Utilização de dificuldade em jogos	17
2.1.1 Dificuldade adaptativa	18
2.2 Flow e experiência percebida em atividades	19
2.3 Representação de habilidade de jogadores	20
2.4 Progressão em Jogos	21
2.5 Tipos de jogadores	22
2.6 Utilização de grafos em jogos	23
2.7 Classificação de jogos por sua topologia	24
2.8 Algoritmos de navegação	26
2.9 Considerações Parciais	27
3 Trabalhos relacionados	28
3.1 Definição dos trabalhos	28
3.2 Implementing Adaptive Game Difficulty Balancing in Serious Games (2018)	28
3.3 Dynamic Difficulty Adjustment Through an Adaptive AI (2015)	29
3.4 Performance-Based Dynamic Difficulty Adjustment and Player Experience	30
3.5 Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review (2018)	30
3.6 Boredom, Engagement and Anxiety as Indicators ... in Games (2008) . . .	31
3.7 Dynamic difficulty adjustment for maximized engagement in digital games (2017)	31
3.8 Conclusões Parciais	33
4 Desenvolvimento	35
4.1 Representação de desafios e objetivos de jogos em grafos	35
4.2 Estrutura de dados para os nós e arestas	36
4.3 Visão geral do simulador	37
4.4 Parâmetros da Simulação	39
4.4.1 Início e fim da simulação	39
4.4.2 Habilidade do agente	39
4.4.3 Número de iterações	40

4.4.4	Modelo de Perfil de Jogador	40
4.4.5	Modelos de desafios	41
4.4.6	Modelo de desistência (<i>churn</i>)	42
4.4.7	Modelo de progressão do jogador	43
4.4.8	Modelo de dificuldade	45
4.5	Estrutura e funcionalidades do <i>Level Designer Planning and Simulation Tool</i> (SIMUPRO)	46
5	Avaliação de Cenários	49
5.1	Cenário de referência	49
5.2	Cenário 1: Modelo de desafio por habilidade	50
5.3	Cenário 2: Modelo de progressão de habilidade	52
5.4	Cenário 3: Modelo de perfil de jogador explorador	53
5.5	Cenário 4: Modelo de desistência por <i>flow</i> (tédio)	53
5.6	Cenário 5: Modelo de desistência por <i>flow</i> (ansiedade)	54
5.7	Cenário 6: Topologia em grade	56
5.8	Cenário 7: Topologia de rede	57
5.9	Cenário 8: Aplicação de um ajuste manual	58
5.10	Cenário 9: Ajuste de dificuldade adaptativo	60
5.11	Cenário 10: Ajuste de dificuldade adaptativo (2)	62
5.12	Considerações Parciais	64
6	Considerações Finais	65
	Bibliografia	67

Lista de Figuras

2.1	A zona de <i>flow</i> representada em branco e os extremos representados pela região de tédio em azul e a região de estresse em amarelo.	20
2.2	Representação de um grafo completo, conexo e planar com quatro vértices e seis arestas.	23
2.3	Mapa do jogo <i>HarmoKnight</i> representado por nós e arestas.	24
2.4	Topologia dos diferentes tipos de jogos em grafos.	25
2.5	Representação da busca em profundidade pela sequência dos números nos nós.	26
2.6	Representação da busca A* achando o menor caminho do nó A para o nó C	27
3.1	Modelo de nível e tentativa em grafo.	32
4.1	Representação de desafios de um jogo em grafos.	36
4.2	Representação da estrutura do nó onde as propriedades do SIMUPRO estão na direita com a posição do nó e as propriedades do <i>CytoscapeJS</i> na esquerda.	37
4.3	Representação da estrutura da aresta onde as propriedades do SIMUPRO estão à direita com a posição do nó e as propriedades do <i>CytoscapeJS</i> à esquerda.	37
4.4	Ilustração do processo de utilização do SIMUPRO. O usuário preenche os parâmetros de simulação e junto com o grafo criado pelo mesmo gera os dados que podem ser observados pelos mapas de calor. As cores frias são usadas para representar valores pequenos e tons quentes para valores altos	38
4.5	Representação gráfica do início e fim do jogo. O nó inicial é identificado pelo número 10 e é destacado com formato de um quadrado, simbolizando o ponto de partida do jogo. Já o nó final, representado pelo número 6, está destacado com o formato de um hexágono, indicando o objetivo a ser alcançado.	39
4.6	Captura de tela do SIMUPRO. Temos os três componentes separados e designados por letras. Temos o painel de configuração designado pela letra “A”, o editor de topologia pela letra “B” e por último, as ferramentas de edição e visualização pela letra “C”. Fonte: do autor.	48
5.1	Modelagem e resultado da simulação do cenário de referência. Fonte: do autor.	50
5.2	Cenário 1: modelo de desafio por habilidade usando o Elo <i>rating</i> . Fonte: do autor.	51
5.3	Mapa de calor depois de 10.000 iterações simuladas do Cenário 2 habilitando o modelo de progressão de jogador. Fonte: do autor.	52
5.4	Mapa de calor para o Cenário 3 com perfil de jogador explorador que realiza uma busca em profundidade. Fonte: do autor.	53
5.5	Representação gráfica do Cenário 4 com modelo de desistência (<i>churn</i>) por tédio.	55
5.6	Representação gráfica do Cenário 5 com os resultados representados por um mapa de calor. Fonte: do autor.	56

5.7	Representação gráfica do Cenário 6 com modelo de chance e perfil realizador. Fonte: do autor.	57
5.8	Representação gráfica do Cenário 7 com os resultados representados por um mapa de calor. Fonte: do autor.	58
5.9	Representação gráfica do Cenário 8 com os resultados em mapas de calor. Fonte: do autor.	59
5.10	Representação gráfica do Cenário 9 com os resultados em mapas de calor. Fonte: do autor.	62
5.11	Representação gráfica do Cenário 10 com os resultados representados por um mapa de calor. Fonte: do autor.	63

Lista de Abreviações e Siglas

MOBA *Multiplayer Online Battle Arena*. 33

SIMUPRO *Level Designer Planning and Simulation Tool*. 7, 8, 16, 34–39, 46–49, 56, 64

SVM Support Vector Machine. 31

1 Introdução

1.1 Apresentação do Tema

A influência dos videogames em nossa sociedade é cada vez mais evidente. Isso pode ser observado pelo sucesso alcançado no lançamento de inúmeros filmes baseados em jogos, assim como pela crescente presença dos jogos na cultura popular (PACETE, 2022). Os jogos deixaram de ser meramente uma forma de entretenimento e passaram a integrar nossa cultura por meio de brinquedos, colecionáveis e até mesmo vestuário (HERINGER, 2022).

Com o crescente reconhecimento e aceitação dos jogos na sociedade, as empresas estão investindo cada vez mais recursos no desenvolvimento de personagens, mecânicas e mundos virtuais cada vez mais complexos. Essa busca por jogos de qualidade resulta na formação de comunidades virtuais massivas, em que os jogadores se reúnem para compartilhar suas experiências e discutir sobre os jogos. As empresas, ao conseguirem criar uma comunidade virtual engajada, têm maiores chances de manter a atenção dos usuários e, conseqüentemente, gerar mais lucro com a venda de jogos e produtos colecionáveis.

1.2 Contextualização

O estabelecimento de empresas como Sega, Nintendo e Atari no ramo dos jogos trouxe o renascimento dos fliperamas, impulsionado por jogos icônicos como Pong (1972), Pac-Man (1980) e Space Invaders (1978) (KENT, 2010). Esses fliperamas se tornaram locais atrativos para pessoas com dificuldades em socializar, oferecendo uma chance de obter admiração por meio do sucesso nos jogos (WILLIAMS, 2006).

No entanto, a dificuldade dos jogos nessas décadas era determinada principalmente pelo tempo de jogo e pela quantidade de moedas gastos, visando evitar que os jogadores monopolizassem uma máquina e também garantir que os jogos não fossem muito difíceis a ponto de serem impossíveis de serem jogados (LOWOOD; GUINS, 2016).

Esse enfoque monetário para definir a dificuldade não considerava o contexto do jogo, a acessibilidade e o design, o que poderia deixar os jogadores insatisfeitos, uma vez que o nível de habilidade não era considerado (LOWOOD; GUINS, 2016).

Atualmente, existem diversas abordagens para aplicar a dificuldade nos jogos e alterar a experiência do jogador. Os desafios são criados com uma dificuldade considerando o contexto, história do jogo, e a vários elementos, como personagens, ambiente e estética. Isso significa que a dificuldade pode influenciar aspectos como a acessibilidade, permitindo que jogadores com diferentes habilidades possam desfrutar do jogo, e a estética, adaptando a interface do jogo conforme a dificuldade (LOWOOD; GUINS, 2016).

1.3 Descrição do Problema

Um dos elementos que contribui para a diversão em um jogo é o desafio que ele apresenta. O desafio pode se manifestar de várias formas, mas precisa ter um objetivo claro para que o jogador possa avaliar seu progresso em direção a ele. Além disso, o jogo deve proporcionar um ambiente que desperte a curiosidade, equilibrado entre complexidade e simplicidade para garantir o sucesso da experiência do jogador (MALONE, 1980).

No entanto, surge um problema quando se trata de determinar a complexidade dos ambientes e a dificuldade dos jogos, pois nem sempre eles atendem às necessidades de todos os jogadores. Considerando que as pessoas possuem habilidades cognitivas e motoras diferentes, surge a necessidade de um sistema de dificuldade que possa fornecer uma experiência equivalente para todos os jogadores (SILVA; SILVA; CHAIMOWICZ, 2015).

Por esse motivo, alguns dos jogos contemporâneos estão adotando a dificuldade adaptativa para manter os desafios alinhados com o desempenho de cada jogador, permitindo que todos possam desfrutá-lo, independentemente de sua experiência prévia (SILVA; SILVA; CHAIMOWICZ, 2015).

Neste trabalho, abordamos o problema mencionado apresentando uma ferramenta para simular como diferentes tipos de jogadores se comportam em diferentes ambientes de jogos. O projeto proposto visa representar diferentes ambientes de jogos, para

simular tipos de jogadores e modelos de dificuldade por uma ferramenta de simulação. Espera-se aumentar o conhecimento sobre as diferentes abordagens de técnicas de dificuldade adaptativa pela avaliação dos dados obtidos por simulações. Com a grande variação de heurísticas e representações é esperado que o trabalho sirva como base para outras pesquisas relacionadas ao tema.

1.4 Justificativa e Motivação

O engajamento do jogador requer um equilíbrio entre o desafio apresentado e a recompensa esperada, de modo que o jogador sinta que mereceu a recompensa (HENDRIX et al., 2018; CSIKSZENTMIHALYI, 1990). Jogadores mais experientes e familiarizados com o gênero do jogo podem ficar desmotivados ao completarem o jogo com muita facilidade. Por isso, os jogos estão cada vez mais adotando a dificuldade adaptativa, ajustando-a com base no estilo de jogo da pessoa, por exemplo, alterando a disponibilidade de itens conforme o desempenho do jogador. Essa abordagem tem recebido apoio da comunidade e demonstrado aumentar o engajamento e a retenção dos jogadores (DENISOVA; CAIRNS, 2019).

Pesquisas como as de Zohaib (2018) e Robb e Zhang (2022) destacam os benefícios da dificuldade adaptativa em manter os jogadores engajados. Além disso, estudos como o de Hendrix et al. (2018), que propôs um plano para aplicar a dificuldade adaptativa em jogos, ressaltam a necessidade de uma ferramenta global que possibilite a simulação e representação da dificuldade adaptativa em diferentes ambientes.

Embora as técnicas de dificuldade adaptativa tenham mostrado sua importância na incorporação de conceitos complexos em sistemas dinâmicos, ainda há oportunidades para representar a dificuldade de maneiras diferentes nos jogos. A criação de uma ferramenta global que facilite a simulação da dificuldade adaptativa em diferentes gêneros de jogos também é uma área promissora (ZOHAI, 2018).

1.5 Questões de Pesquisa e Hipótese

No contexto da dificuldade adaptativa em jogos, como discutido nas seções anteriores, este trabalho busca abordar questões críticas por meio de uma análise da literatura e simulações. Primeiramente, como a dificuldade de um jogo pode ser medida e modelada, levantando a seguinte questão: “Como podemos modelar o progresso de dificuldade jogos” A compreensão desse aspecto é fundamental para abordar adequadamente a adaptação da dificuldade de encontros com base nas ações do jogador, a fim de mantê-lo engajado. Essa questão nos leva a explorar modelos de jogador e sua conexão com o modelo de dificuldade.

Dessa forma, surge a hipótese deste trabalho: um ambiente integrado de modelagem e simulação gráfica de progressão de dificuldade melhorará a compreensão do assunto. Esse artefato resultante pode ser aplicado tanto em um contexto educacional, para transmitir o conhecimento da literatura, quanto para possibilitar a criação de novos modelos de dificuldade adaptativa.

Ao abordar essas questões críticas e desenvolver um ambiente de simulação eficaz, esperamos avançar no campo da dificuldade adaptativa em jogos, proporcionando uma base sólida para futuras pesquisas e aprimorando a compreensão geral desse importante aspecto do design de jogos.

1.6 Objetivos

Esse projeto visa aumentar o entendimento sobre diferentes abordagens de representação da dificuldade adaptativa em level design, por meio da criação de uma ferramenta para simular diferentes ambientes de jogos. Essa ferramenta tem como objetivo facilitar a manipulação da dificuldade por designers e desenvolvedores em empresas, contribuindo para o processo de desenvolvimento dos jogos.

Para sua realização, temos os seguintes objetivos específicos:

- Possibilitar a edição de topologia de grafos para representar a estrutura do ambiente de jogos;

- Simular o percurso realizado pelos jogadores nos jogos utilizando algoritmos de busca;
- Permitir a configuração do simulador para adaptar o ambiente e suas características;
- Criar um modelo de adaptação de dificuldade baseado nas ações do jogador simulado;
- Possibilitar a análise dos dados nos grafos por meio de mapas de calor;

Esses objetivos específicos têm como finalidade criar uma base sólida para o desenvolvimento da ferramenta, aumentando a variedade de abordagens possíveis para diferentes gêneros de jogos. A configuração flexível da ferramenta será desenvolvida para facilitar a sua utilização, permitindo que o usuário manipule as variáveis de acordo com suas necessidades, a fim de obter os resultados desejados.

1.7 Metodologia

Inicialmente, foi realizado um estudo sobre o uso de dificuldade adaptativa em jogos para melhorar o engajamento dos jogadores. Esse estudo demonstrou que existem inúmeras maneiras eficientes de ajustar a dificuldade de um jogo para melhorar métricas de engajamento, como a quantidade de vezes que um jogo foi acessado e a duração de uma sessão de jogos (ZOHAI, 2018). Analisando os resultados obtidos nas pesquisas relacionadas foi possível perceber a eficácia que os sistemas de dificuldade adaptativo tem em manter o usuário na zona de *flow* (CSIKSZENTMIHALYI, 1990).

Após o estudo da literatura foi notada uma variação nas metodologias utilizadas e na maneira que a dificuldade era medida. Além disso, as pesquisas estudadas tinham em mente um escopo específico onde o contexto estava ligado a um determinado gênero de jogo. Tendo isso em mente, neste projeto foi especificada uma abordagem mais global de dificuldade adaptativa que a partir de um conjunto de variáveis de entrada em um grafo determina a melhor maneira de manipular a dificuldade em um ambiente para melhorar o engajamento dos jogadores.

Neste trabalho foi desenvolvido um ambiente de simulação chamado SIMUPRO, no qual conjuntos de parâmetros como o tipo de jogador, dificuldade, desistência, progressão de habilidade e desafio são utilizados para simular a navegação pelo conteúdo do jogo preparado pelo *game designer* na forma de um nível ou mesmo do jogo como todo, em diversos ambientes.

O ambiente de simulação foi implementado utilizando a linguagem *TypeScript*, com o intuito de apresentar um ambiente onde o responsável pelo planejamento do jogo pode criar uma rede representando os diversos caminhos do jogador e o grau de desafio para essa navegação. A biblioteca de grafos *CytoscapeJS* foi utilizada para a representação na forma de grafos. A base do projeto foi feita com a biblioteca *React* visando apresentar uma interface gráfica visualmente agradável para o usuário e facilitar a realização de testes e manutenção, bem como seu acesso pela *web*.

Após a construção da ferramenta, foi realizado um estudo sobre os possíveis tipos de combinação dos parâmetros utilizados para representar diferentes cenários de simulação. As topologias dos grafos, os parâmetros de simulação e resultados obtidos foram avaliados para determinar a melhor técnica aplicada comparando os resultados obtidos pelas métricas escolhidas. A comparação dos dados obtidos foi realizada por mapas de calores configuráveis gerados pela simulação dos cenários.

1.8 Organização do texto

O texto está estruturado em seis capítulos, cada um abordando aspectos específicos. Após esta Introdução, o Capítulo 2 fornece uma base teórica do projeto, incluindo uma revisão bibliográfica de trabalhos acadêmicos, científicos e da indústria. Em seguida, o Capítulo 3 detalha os trabalhos relacionados usados como base e realiza uma breve comparação entre eles. O Capítulo 4 descreve o conjunto de modelos, equações e algoritmos implementados na ferramenta SIMUPRO aqui desenvolvido e o Capítulo 5 analisa as diversas combinações de modelos, parâmetros e resultados das simulações. Por fim, no Capítulo 6 temos as considerações finais deste trabalho, bem como limitações e sugestões de trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo os principais conceitos necessários para o entendimento deste trabalho são discutidos.

2.1 Utilização de dificuldade em jogos

Uma das definições mais populares, preferidas do famoso *game designer* Sid Meier, é que um jogo é uma série de escolhas interessantes. Essas escolhas ocorrem em desafios autoimpostos, podendo ser fáceis ou difíceis para o jogador, com base em sua habilidade mental ou motora e no projeto realizado pelo *game designer*. Um dos principais fatores que determinam a qualidade de um jogo é a presença de um sistema de dificuldade que desafie adequadamente o jogador durante a sua experiência (APONTE; LEVIEUX; NATKIN, 2009). Além disso, a satisfação de jogar um jogo está relacionada à incerteza do resultado, ou seja, um jogo se torna entediante quando o jogador tem certeza de que irá ganhar ou perder (MALONE, 1980).

Uma forma de manter a ambiguidade do resultado para jogadores com diferentes habilidades é por meio de níveis de dificuldade variáveis, que podem ser classificados em três tipos: dificuldade selecionada pelo usuário a partir de opções disponíveis no jogo; dificuldade baseada na habilidade do oponente(jogadores); e dificuldade definida pelo programa com base no desempenho do jogador (MALONE, 1980).

A empresa Atari popularizou jogos em que o jogador pode escolher o nível de dificuldade, seguindo o princípio do fundador da empresa, Nolan Bushnell, de que “os melhores jogos são fáceis de aprender e difíceis de dominar”. Jogos como *Tempest* (Atari, 1981), *Star Wars* (Atari, 1983) e *RoadBlasters* (Atari, 1987) apresentavam um menu de configurações em que o usuário podia escolher a dificuldade do jogo.

A oportunidade para o jogador de decidir o nível de desafio que melhor se adapta às suas habilidades resultou em um maior engajamento (ARAUJO; FEIJÓ, 2013). A partir disso, os jogos passaram a oferecer um sistema de dificuldade com opções como

fácil, normal e difícil (LOWOOD; GUINS, 2016).

Apesar de melhorar o engajamento dos jogadores, jogos com uma seleção de dificuldade pré-definida proporcionam uma experiência estática e podem levar a resultados previsíveis, uma vez que os níveis de dificuldade apresentados são subjetivos para cada pessoa (WONG et al., 2006). Além disso, criar uma curva de dificuldade bem equilibrada, aumentando gradualmente o desafio de um jogo de maneira satisfatória, é um dos principais desafios no design de jogos (APONTE; LEVIEUX; NATKIN, 2009).

Mesmo com designers experientes desenvolvendo a curva de dificuldade, ainda existem diversos problemas que não podem ser abordados completamente. Problemas como a diversidade entre os jogadores, suas experiências prévias, habilidades individuais, abordagens aos problemas e adaptabilidade a desafios específicos. Além disso, as preferências de dificuldade de um jogador podem mudar ao longo do tempo (XUE et al., 2017).

2.1.1 Dificuldade adaptativa

Considerando os problemas dos jogos com dificuldade estática mencionados anteriormente, outra abordagem para definir os desafios é com a utilização de uma dificuldade adaptativa. Dessa maneira, a dificuldade dos desafios se ajusta automaticamente às habilidades do jogador ao longo do jogo, para manter a dificuldade ideal, reduzindo a frustração e o tédio (ZOHAIB, 2018). Esse ajuste pode ser realizado conforme o desempenho do jogador em jogos contra monstros ou de acordo a habilidade de outros jogadores.

Para ilustrar o conceito de jogos contra monstros, podemos citar o jogo *Left 4 Dead* (Valve Corporation, 2008). Nesse jogo, é utilizado um sistema inteligente que determina a quantidade de inimigos e desafios para manter o ritmo e a intensidade da experiência (LOWOOD; GUINS, 2016).

A dificuldade adaptativa também é amplamente aplicada no contexto do *matchmaking*, ou pareamento, em jogos multijogadores, visando proporcionar duelos equilibrados entre eles. O *matchmaking* é o processo de selecionar adversários ou parceiros de jogo com base em suas habilidades, experiência e nível de desafio desejado. Ao utilizar a dificuldade adaptativa nesse contexto, os sistemas de *matchmaking* são capazes de ajustar

as partidas para garantir que os jogadores enfrentem oponentes com um nível de habilidade semelhante, criando uma competição justa e emocionante (BALDWIN; JOHNSON; WYETH, 2014).

2.2 Flow e experiência percebida em atividades

Na Subseção 2.1.1, vimos que a dificuldade adaptativa é uma abordagem utilizada para ajustar os desafios de um jogo para manter uma equivalência entre a habilidade dos jogadores e os desafios apresentados. Essa abordagem está relacionada ao conceito de *flow* (CHEN, 2007). O conceito de *flow*, foi introduzido por Csikszentmihalyi (1990) em 1970 para descrever a sensação de felicidade. Ele explicou que no estado de *flow*, uma pessoa se sente completamente imersa em uma atividade, experimentando prazer e satisfação. O conceito, inicialmente associado a atletas e artistas, passou a representar qualquer atividade humana.

A Figura 2.1 ilustra um gráfico que representa esse conceito: delimitado por uma área branca, temos os eixos vertical e horizontal, que representam o desafio e a habilidade da pessoa, respectivamente. Para alcançar o estado desejado de *flow*, caracterizado por prazer, foco, perda de consciência corpórea e temporal, é necessário haver uma equivalência entre os valores dos dois eixos.

Os extremos do gráfico, representados pelas cores amarela e azul, demonstram uma desigualdade em um dos eixos. A região amarela ocorre quando o desafio apresentado é superior às habilidades do jogador, resultando em frustração por ansiedade, levando à desistência. A região azul ocorre quando o desafio é muito fácil para a pessoa, causando tédio e desinteresse.

Para uma atividade ou tecnologia atingir a zona de *flow*, é preciso satisfazer pelo menos alguns dos componentes a seguir:

- Enxergar as atividades como um desafio;
- Acabar com a procrastinação;
- Realizar uma atividade de cada vez;

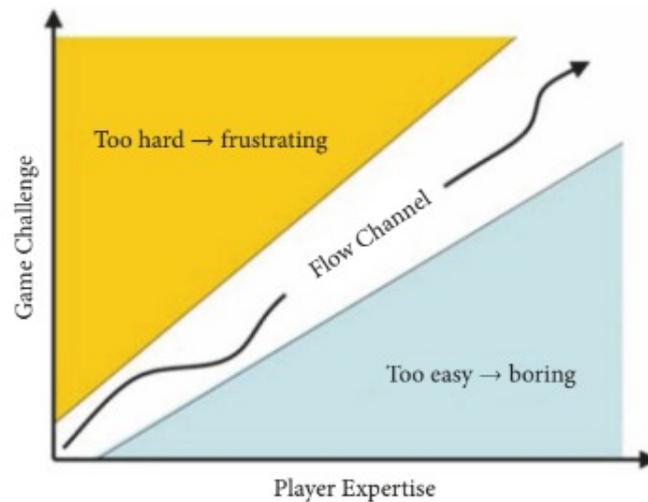


Figura 2.1: A zona de *flow* representada em branco e os extremos representados pela região de tédio em azul e a região de estresse em amarelo. Fonte: Zohaib (2018).

- Fazer o que você ama;
- Evitar a autocrítica excessiva;
- Estar no ambiente certo;
- Desenvolver o autoconhecimento.

Ao analisar esses componentes, percebemos que a dificuldade adaptativa em jogos está diretamente relacionada ao estado de *flow* e à imersão do jogador (KOSTER, 2013; CHEN, 2007). Visto que alguns dos componentes citados s'ao aplicado em jogos. Além disso,

Ao analisar esses componentes, fica evidente a conexão entre a dificuldade adaptativa nos jogos e a experiência de *flow*, caracterizada pela imersão do jogador (KOSTER, 2013; CHEN, 2007). É importante destacar que diversos desses componentes têm aplicação direta na concepção e no design de jogos. Além disso, para que um jogo seja realmente satisfatório, como já vimos, é necessário encontrar o equilíbrio perfeito entre o desafio proposto e as habilidades do jogador (MALONE, 1980)

2.3 Representação de habilidade de jogadores

Para essa pesquisa, o sistema Elo *rating* foi utilizado para classificar a habilidade dos jogadores e a dificuldade dos desafios. O Elo *rating* é um sistema amplamente empregado

para classificar jogadores em esportes competitivos, originalmente desenvolvido para o xadrez e posteriormente adaptado para diversos contextos, incluindo jogos digitais.

A essência do sistema Elo *rating* é atribuir a cada jogador uma pontuação inicial e, a partir dos resultados de suas partidas, ajustar essa pontuação. Quando um jogador vence um adversário com uma pontuação mais alta, ele ganha mais pontos em comparação a uma vitória contra um oponente com pontuação semelhante ou inferior (ELO, 1978).

2.4 Progressão em Jogos

A progressão em jogos refere-se à maneira como os jogadores evoluem e avançam no jogo ao longo do tempo (BJÖRK; HOLOPAINEN, 2005). A maneira que a mesma é aplicada em um jogo é muito importante para a experiência do jogador, visto que é uma das cinco funções básicas dos jogos, juntamente com o desafio, a sorte, a fantasia e o companheirismo (KOSTER, 2013). Tendo isso em mente, desenvolvedores de jogos a classificam em quatro tipos: progressão do personagem; progressão por habilidade; progressão da narrativa; e progressão pela mecânica do jogo.

A progressão do personagem é a evolução do personagem controlado ao longo do jogo, seja em termos de habilidades, estatísticas, equipamentos ou outras características (SCHREIBER; ROMERO, 2021). Já a progressão por habilidade é realizada pela melhoria da capacidade do jogador, ou seja, a evolução da resolução de problemas, coordenação e tomada de decisões. A progressão da narrativa do jogo é a progressão da história do jogo, incluindo a apresentação de enredos, personagens, conflitos e desafios desenvolvidos ao longo do tempo (SALEN; ZIMMERMAN, 2004). Por último, temos a progressão da mecânica do jogo que envolve o desenvolvimento e a introdução de novas mecânicas de jogo ao longo do tempo, como a introdução de novos elementos de jogabilidade, a variação de níveis de dificuldade e a adição de novos obstáculos (HUNICKE; LEBLANC; ZUBEK, 2004).

2.5 Tipos de jogadores

Como vimos na Seção 2.4, a progressão de um jogo pode estar relacionada à habilidade e experiência do jogador. Além disso, na Seção 2.2 foi destacado que, para um jogo ser satisfatório, é necessário que o desafio proposto esteja em equilíbrio com as habilidades do jogador, permitindo a entrada na zona de *flow*.

Com isso em mente, neste trabalho, foi adotada uma nomenclatura para definir os tipos de jogadores para melhor representá-los. A taxonomia de Bartle divide os jogadores em quatro tipos principais: exploradores; realizadores; socializadores; e combatentes (BARTLE, 1996).

Os jogadores do tipo exploradores têm interesse em explorar o mundo do jogo, descobrir mecânicas novas, experimentar e aprender mais sobre o jogo. Eles exploram áreas ocultas que não estão relacionadas à história principal, a fim de descobrir tudo o que o jogo tem a oferecer.

Os socializadores têm interesse em interagir com outros jogadores e criar relacionamentos na comunidade do jogo. Eles buscam compartilhar experiências e buscam atividades que tem interações com outros jogadores.

Semelhante aos jogadores exploradores, os realizadores também exploram diversos aspectos do jogo, mas eles são motivados por metas e conquistas, e tendem a se concentrar em concluir objetivos específicos no jogo, como completar a história principal.

Os jogadores do tipo combatentes se concentram em derrotar inimigos. Eles podem se divertir ao enfrentar inimigos desafiadores e descobrir novas maneiras de vencer batalhas difíceis. Esses tipos de jogadores podem ser movidos pela emoção da competição, pela satisfação de superar obstáculos e pela busca de novas experiências no jogo (YEE, 2006).

Neste trabalho, os tipos exploradores e realizadores foram modelados para guiar a progressão da narrativa no jogo.

2.6 Utilização de grafos em jogos

Um grafo é uma abstração matemática que consiste em um par de conjuntos. O primeiro é um conjunto não vazio $V(G)$ que representa os vértices do grafo, enquanto o segundo conjunto $A(G)$ são pares distintos não ordenados de elementos distintos de $V(G)$, representando as arestas (COSTA, 2011). Em outras palavras, os vértices são os pontos do grafo e as arestas são as conexões entre esses pontos.

Na Figura 2.2 temos um grafo conexo. Através dele é possível visualizar como os vértices (d, a, b, c) e as arestas (linhas em preto) interagem e podem ser utilizados para representar pontos e caminhos. Essa representação pode ser aplicada em diversos problemas, como, por exemplo, o problema do caixeiro viajante, onde busca-se determinar, a partir de uma cidade (vértice), a melhor maneira de visitar todas as outras cidades e retornar para a cidade inicial com a menor distância possível (COSTA, 2011; FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011).

Além deste problema, os grafos também podem ser usados para resolver o problema do menor caminho entre dois pontos. Nesse caso, o objetivo é encontrar a rota mais curta entre dois vértices específicos no grafo. Isso pode ser especialmente relevante em situações como planejamento de rotas, navegação, redes de transporte, entre outros (COSTA, 2011).

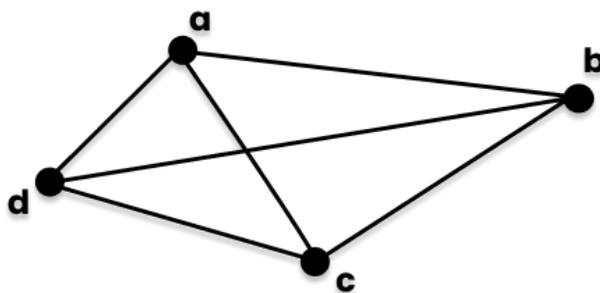


Figura 2.2: Representação de um grafo completo, conexo e planar com quatro vértices e seis arestas. Fonte: (COSTA, 2011).

Tendo em mente os problemas vistos onde um grafo é aplicado, podemos usar o mesmo para aplicações e problemas similares em jogos. Os vértices podem representar os objetivos de um jogo e as arestas os desafios entre esses objetivos. A Figura 2.3 ilustra um tipo de representação onde a progressão da narrativa de um jogo é representada por

um grafo, onde os vértices correspondem a níveis no jogo e as arestas representam as possíveis conexões entre esses níveis.

Além da figura citada como um exemplo, pesquisas já foram conduzidas na parte de jogos por grafos. O trabalho de Xue et al. (2017) adotou uma representação bidimensional do espaço dos vértices, como nível e a tentativa em um jogo, enquanto as transições dos jogadores foram representadas como arestas, refletindo a movimentação para um novo nível ou tentativa, conforme o contexto. Essa abordagem demonstra a capacidade dos grafos em representar a dificuldade no espaço e sua utilidade para modelar a navegação em caminhos específicos.

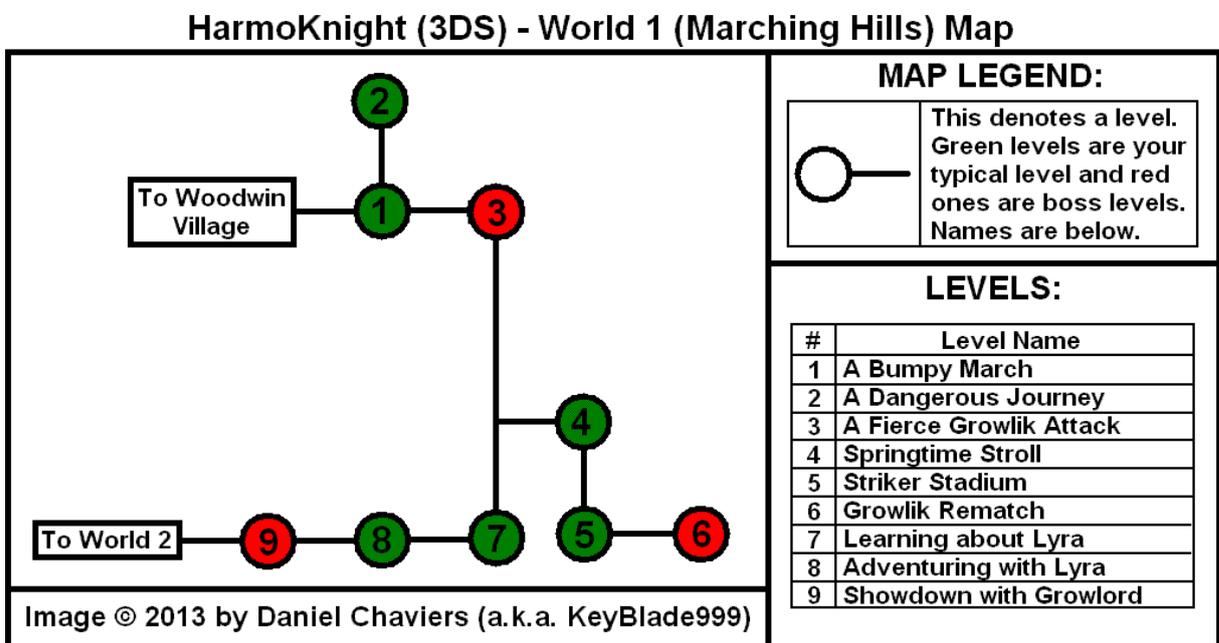


Figura 2.3: Mapa do jogo *HarmoKnight* representado por nós e arestas. Fonte: VGmaps (<https://www.vgmaps.com/Atlas/3DS/HarmoKnight-World1-MarchingHills.png>)

2.7 Classificação de jogos por sua topologia

Tendo em mente a possível utilização dos grafos em jogos, como discutido na Seção 2.6, vimos que é possível utilizar a estrutura do mesmo para representar o conceito de menor caminho entre dois pontos. Com isso em mente, a classificação dos jogos nesta pesquisa é baseada na topologia dos grafos, ou seja, na forma como o jogador simulado, representado pelos algoritmos de busca, pode navegar de um vértice para outro. É importante ressaltar que, nesse contexto, a topologia não se refere a um grafo topológico específico, mas sim à

organização dos vértices e arestas que compõem o ambiente de jogo.

Neste trabalho a classificação dos jogos é baseada na topologia dos grafos e foi definida em três tipos distintos: jogos lineares; jogos de grade; e jogos de rede, conforme pode ser visto na Figura 2.4.

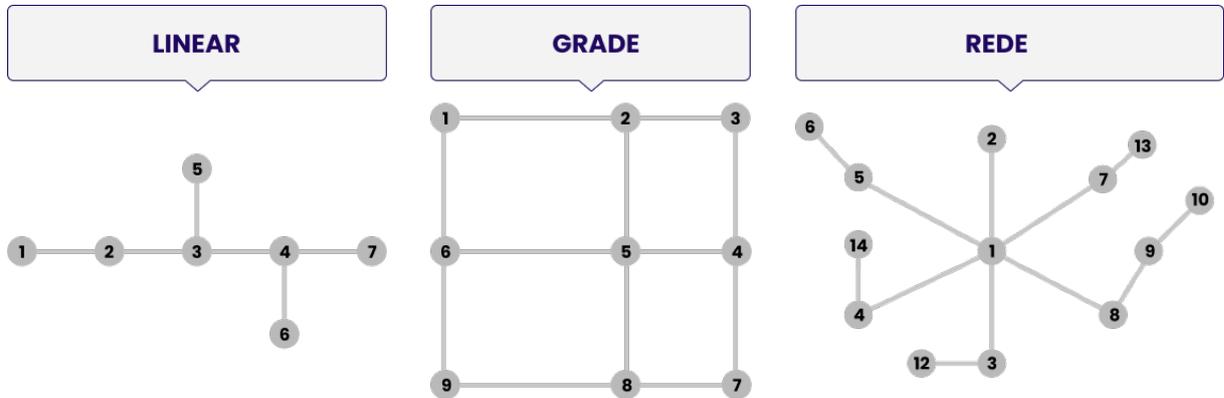


Figura 2.4: Topologia dos diferentes tipos de jogos em grafos. Fonte: do autor.

Os jogos lineares neste trabalho seguem a nomenclatura de Schell (2008), onde o jogador segue um caminho predeterminado, sem muitas escolhas ou ramificações, podendo ser representados por uma série de desafios sequenciais. No grafo, esse tipo de jogo tem uma quantidade de nós e arestas reduzidas, diminuindo as possíveis escolhas do jogador. Alguns exemplos de jogos lineares incluem Super Mario Bros. Deluxe (NINTENDO, 2019) e Rayman Legends (ANCEL; HÉRAL; MONTPELLIER, 2013).

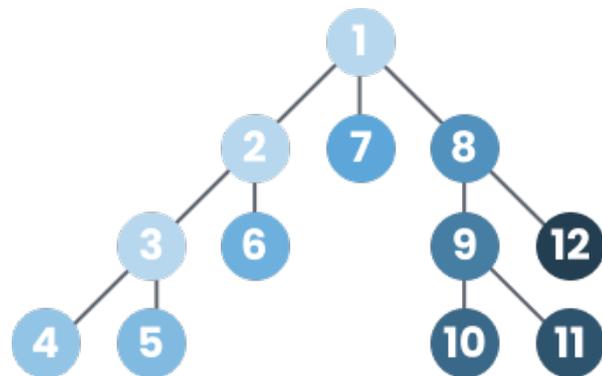
Os jogos do tipo grade são representados neste projeto pela nomenclatura de Björk e Holopainen (2005) e utilizam uma grade de células bidimensional para representar o mundo do jogo, em que cada célula da grade representa uma posição no espaço do jogo e os objetos do jogo são posicionados nessas células. Exemplos de jogos do tipo grade incluem jogos de tabuleiro, como xadrez e o Super Bomberman 5 (Konami, 1997). No grafo eles apresentam uma estrutura mais compacta com uma abundância de ligação entre os nós.

Por último, os jogos do tipo rede também usam a nomenclatura de Björk e Holopainen (2005). Nesses jogos, os jogadores podem escolher livremente seus caminhos e objetivos. Exemplos de jogos do tipo rede incluem jogos de simulação, como o The Legend of Zelda: A Link to the Past (Nintendo, 1991) e Dark Souls (FromSoftware, 2011). No grafo, esses jogos apresentam vários caminhos e podem ser muito extensos.

2.8 Algoritmos de navegação

Na Seção 2.6 vimos que é possível representar o progresso de um jogador em um jogo por um grafo pelos vértices e arestas. Com a representação de caminhos e pontos no grafo, podemos buscar a melhor forma de navegar entre os vértices. Levando isso em consideração, para o desenvolvimento deste trabalho, foram utilizados dois algoritmos de busca: busca em profundidade; e a busca A* (A estrela). Estes algoritmos serão brevemente definidos com base nos trabalhos de (CORMEN et al., 2009; SEDGEWICK; WAYNE, 2011)

A busca em profundidade é um algoritmo de busca em grafos que explora o máximo possível de um ramo do grafo antes de retroceder. Começando por um vértice inicial, o algoritmo visita e processa cada vértice encontrado, marcando-os como visitados. Em seguida, ele busca por vértices adjacentes não visitados e continua o processo de forma recursiva. O algoritmo retrocede somente quando encontra um vértice sem filhos. A Figura 2.5 ilustra o funcionamento do algoritmo.



Busca em profundidade

Figura 2.5: Representação da busca em profundidade pela sequência dos números nos nós. Fonte: adaptado de Needham e Hodler (2019).

A busca A* (A estrela) é um algoritmo de busca heurística em grafos que combina o custo do caminho percorrido com uma estimativa heurística do custo para alcançar o objetivo final. Essa estimativa é baseada em uma função heurística que estima a distância entre um determinado vértice e o objetivo final. Na Figura 2.6 temos a demonstração do funcionamento deste algoritmo que busca o menor caminho do nó A para o nó C.

Ao analisarmos os algoritmos de busca descritos, fica evidente que cada um apresenta um método de pesquisa distinto. Considerando os tipos de jogadores vistos na Seção 2.5, podemos associar as buscas a um tipo de jogador. A busca em profundidade têm a vantagem de representar jogadores que se encaixam nos perfis de exploradores e socializadores, visto que esses jogadores exploram o jogo e não buscam terminar o mais rápido possível. Por outro lado, a pesquisa A*, que prioriza o caminho mais curto, é mais adequada para jogadores combatentes e realizadores.

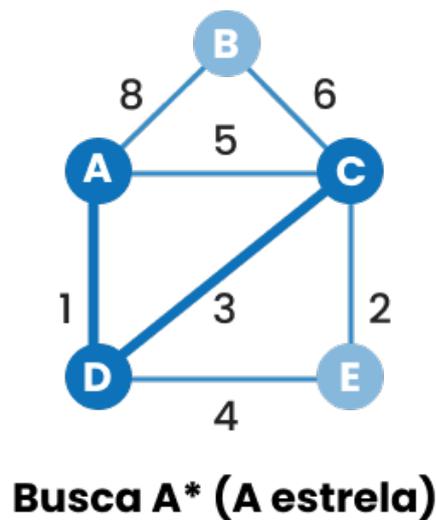


Figura 2.6: Representação da busca A* achando o menor caminho do nó A para o nó C. Fonte: adaptado de Needham e Hodler (2019).

2.9 Considerações Parciais

Neste capítulo, foram explorados os principais conceitos relacionados à dificuldade em jogos utilizados nesta pesquisa. Na Seção 2.1, foi abordada a utilização da dificuldade em jogos. Na Subseção 2.1.1, foi discutida a importância da dificuldade adaptativa. Na Seção 2.2, foi explorado o conceito de *flow* e sua relação com a dificuldade. Nas seções 2.3, 2.4 e 2.5, foi vista a representação da habilidade do jogador, os tipos de progressão e os tipos de jogadores. Nas seções 2.6 e 2.7, foi discutida a utilidade dos grafos na modelagem de jogos e a representação dos tipos de jogos nesses grafos. Na Seção 2.8, foi abordada a representação dos jogadores no grafo. Os conceitos explorados neste capítulo servem como base para o próximo capítulo, que descreve e realiza uma discussão sobre os trabalhos relacionados usados nesta pesquisa.

3 Trabalhos relacionados

Este capítulo faz uma revisão e discussão de um conjunto de trabalhos relacionados ao objetivo geral aqui perseguido. Seus objetivos, métodos e resultados são apresentados e uma comparação final entre eles é realizada.

3.1 Definição dos trabalhos

Durante o período entre 26/04/2023 e 06/07/2023, foi realizada uma pesquisa utilizando o Google Scholar com o objetivo de selecionar trabalhos, tanto em inglês quanto em português, que serviriam como base para este estudo. A busca foi conduzida utilizando os termos “adaptive difficulty” e “flow in games”. Da lista de resultados, o trabalho de Zohaib (2018) apresentou uma revisão das principais pesquisas publicadas sobre o assunto em questão, sendo utilizado como base para a obtenção de mais artigos através do método *snowballing* (STREETON; COOKE; CAMPBELL, 2004).

A aplicação da técnica de dificuldade adaptativa depende da maneira que a dificuldade é representada no jogo. Por esse motivo, o contexto em que a dificuldade está inserida é importante. Tendo isso em mente, para o desenvolvimento deste projeto foram estudados diversos tipos de aplicações de dificuldade adaptativa e analisados os seguintes fatores nas pesquisas: descrição do projeto; implementação do *flow*; adaptação da técnica aplicada; gênero do jogo utilizado; resultados encontrados. Essas cinco variáveis serviram de base para uma análise crítica dos artigos exposta nas seções a seguir.

3.2 Implementing Adaptive Game Difficulty Balancing in Serious Games (2018)

O trabalho de Hendrix et al. (2018) desenvolve um plano de seis etapas para servir como guia para a criação de dificuldade adaptativa em jogos. O plano proposto segue um passo a passo para determinar a melhor maneira de identificar quais variáveis devem ser ajustadas

conforme a teoria do *flow* para impactar na dificuldade. Além disso, os autores buscaram deixar o desenvolvimento o mais simples possível para ser utilizado em diferentes gêneros de jogos, independente da tecnologia e plataforma utilizada.

Para validar o plano proposto, os autores realizaram um estudo de caso em dois tipos de jogo com participantes. O primeiro jogo era do gênero de quebra-cabeça e o segundo um jogo clássico de plataforma. A avaliação do desempenho da dificuldade adaptativa em ambos os jogos foi analisada e demonstrou que a maioria dos participantes sentiram mais prazer nas versões dos jogos com dificuldade adaptativa.

3.3 Dynamic Difficulty Adjustment Through an Adaptive AI (2015)

Existem diversas abordagens para representar a dificuldade em um jogo. Diferente da pesquisa anterior, o trabalho de Silva, Silva e Chaimowicz (2015) apresentou um mecanismo para realizar ajustes dinâmicos da dificuldade em uma partida de um jogo através da manipulação de um sistema de inteligência artificial. Para o desenvolvimento desse projeto foi utilizado o jogo Defence of Ancients (Valve Corporation, 2003). Para determinar a dificuldade ideal na pesquisa poder foram definidos três tipos de modos com suas características: fácil; normal; e difícil.

Foi definida uma função heurística com o objetivo de ajustar a dificuldade do jogo. Essa função levava em consideração a zona de *flow* e avaliava se uma variável específica da função estava dentro do seu limite adequado. Caso a variável não estivesse dentro do limite esperado, o jogo era balanceado para oferecer um desafio adequado. Por meio de simulações, verificou-se que o mecanismo foi capaz de acompanhar de forma considerável as habilidades do jogador, demonstrando que o mecanismo funcionou conforme o esperado.

3.4 Player Experience in a 2D Digital Game: A Controlled Experiment (2022)

As duas pesquisas mencionadas anteriormente basearam-se no modelo de *flow* de Csikszentmihalyi (1990) para realizar ajustes em suas fórmulas. No entanto, essas pesquisas focaram na adaptabilidade da dificuldade do jogo, deixando de lado a análise da experiência do jogador em si. Em contraste, a pesquisa de Robb e Zhang (2022) adota uma abordagem diferente. Utilizando o conceito de *flow*, eles investigaram o aumento do conhecimento sobre a experiência que o jogador vivencia no jogo, incluindo aspectos como ansiedade e tédio.

Nesta pesquisa, foi desenvolvido um jogo digital 2D de meteoros, no qual as variáveis determinavam a velocidade do projétil do jogo. Três tipos de dificuldade foram definidos para realizar uma comparação ao final do experimento: dificuldade adaptativa; dificuldade incremental linear; e dificuldade sem ajuste, que mantinha a mesma velocidade do projétil. O jogo foi apresentado a um grupo de participantes por meio da plataforma Amazon Mechanical Turk ¹. A experiência percebida pelos jogadores foi coletada por meio de um formulário e posteriormente analisada. Os resultados demonstraram que a dificuldade adaptativa foi a que melhor acompanhou o desempenho dos jogadores

3.5 Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review (2018)

No trabalho de Zohaib (2018) é feita uma revisão bibliográfica e análise sobre diversos trabalhos de dificuldade adaptativa. Na pesquisa em questão é descrito o papel da teoria do *flow* (CSIKSZENTMIHALYI, 1990) e como ela é utilizada para desenvolver a dificuldade adaptativa em jogos. Também é definido através das pesquisas estudadas pelo autor o essencial para a criação da dificuldade adaptativa.

Para concluir a pesquisa, o autor citou que a investigação feita demonstrou a quantidade de aplicações diferentes e direções que podem ser representadas pela dificul-

¹<https://www.mturk.com/>

dade adaptativa. O autor também cita que pesquisadores podem ir além de jogos padrões de plataforma aplicando as suas pesquisas a diferentes gêneros de jogos.

3.6 Boredom, Engagement and Anxiety as Indicators for Adaptation to Difficulty in Games (2008)

Para manipular a dificuldade existem diversas heurísticas que podem ser escolhidas. O trabalho de Chanel et al. (2008) utiliza uma abordagem totalmente diferente: a abordagem realiza alterações na dificuldade conforme o reconhecimento de emoções. Para medir as emoções dos participantes, dados de um formulário foram analisados após os mesmos terem participado de uma sessão do jogo Tetris (TETRIS, 1984) em diferentes dificuldades.

Além dos dados obtidos pelo formulário, um Support Vector Machine (SVM) (CORTES; VAPNIK, 1995) foi utilizada para medir os sinais fisiológicos recebidos. Analisando os dados obtidos, o autor confirmou que jogando em diferentes níveis levou o aumento de diferentes estados emocionais e quando a dificuldade manteve ao tédio era o resultado.

Dentre as diversas heurísticas existentes para manipular a dificuldade, o trabalho de Chanel et al. (2008) adota uma abordagem totalmente diferente, ajustando a dificuldade com base no reconhecimento das emoções dos jogadores. Para medir as emoções dos participantes, foram analisados os dados de um formulário preenchido após eles participarem de uma sessão do jogo Tetris em diferentes níveis de dificuldade.

3.7 Dynamic difficulty adjustment for maximized engagement in digital games (2017)

O trabalho de Xue et al. (2017) apresenta uma pesquisa feita em conjunto com a empresa Electronic Arts, em que foi desenvolvido um *framework* de dificuldade adaptativa, com uma otimização global para maximizar o engajamento do jogador ao decorrer do jogo. Utilizando jogos do gênero de *puzzle*, foi criado um modelo do progresso do jogador com um grafo probabilístico onde a dificuldade é manipulada para otimizar as transições de

estado e maximizar o tempo de jogo (Figura 3.1).

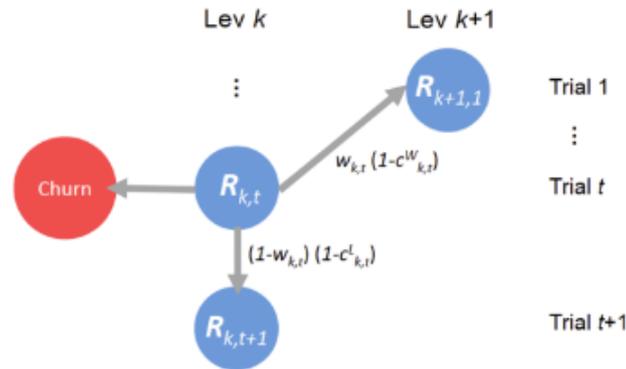


Figura 3.1: Modelo de nível e tentativa em grafo. Fonte: (XUE et al., 2017).

Nesse modelo, denotando o estado do jogador pela tentativa atual representada por t -ésimo *trial* e k -ésimo *Lev* como o nível atual. O usuário consegue avançar, aumentando o seu nível e o número de tentativas com uma transição entre nós ou repete o mesmo nível aumentando somente o número de tentativas. Um dos estados destacado no modelo, representado na Figura 3.1 em vermelho, é chamado de *churn*. Ele representa a desistência do jogador e todos os outros estados podem transitar para ele, representando que a desistência pode ocorrer em qualquer momento.

Com isso, é definida uma função que busca aumentar o engajamento do jogador e diminuir a probabilidade de transições para o estado de desistência *churn*. Para determinar o engajamento dos jogadores na função probabilística foi considerado o número de iterações. O estudo de caso foi realizado nos jogos de quebra-cabeça do mesmo gênero de jogos como Candy Crush Saga (King Digital Entertainment, 2012). Esses tipos de jogo apresentam um sistema de dificuldade gradual onde cada nível aumenta a dificuldade.

Para produzir os ajustes da dificuldade de forma síncrona, o jogo solicitava uma requisição para o sistema de dificuldade adaptativa. Com isso, a dificuldade ideal era calculada para o jogador conforme o estado dele medido pelo progresso e das probabilidades encontradas em cada transição do grafo.

A pesquisa teve sucesso em aplicar a técnica apresentada em diversos jogos da empresa Electronic Arts e teve até 9% de melhoria no engajamento do jogador com um impacto neutro na monetização.

3.8 Conclusões Parciais

A revisão dos trabalhos relacionados revelou diversas abordagens para a implementação da dificuldade adaptativa em jogos. Cada pesquisa apresentou métodos e técnicas específicas, mas todas compartilham o objetivo de melhorar a experiência do jogador por meio do ajuste dinâmico da dificuldade.

Entre a descrição e análise das diversas técnicas de aplicação da dificuldade adaptativa, a comparação realizada na Tabela 3.1 destaca a variedade de jogos e abordagens utilizadas nas pesquisas.

Tabela 3.1: Tabela comparativa entre os projetos estudados.

Nome do trabalho	Gênero do jogo	Abordagem utilizada
Xue et al. (2017)	Puzzle	Modelo de grafos
Zohaib (2018)	—	<i>Survey</i>
Silva, Silva e Chaimowicz (2015)	<i>Multiplayer Online Battle Arena</i> (MOBA)	Inteligência artificial
Hendrix et al. (2018)	Puzzle e plataforma	heurísticas
Robb e Zhang (2022)	<i>Shoot'em up</i>	heurísticas
Chanel et al. (2008)	plataforma	Máquina de vetores de suporte
Este projeto	Independente do Jogo	Modelo de grafos

A revisão dos trabalhos relacionados mostrou diferentes abordagens para a implementação da dificuldade adaptativa em jogos. O trabalho de Hendrix et al. (2018) propôs um plano de seis etapas para guiar a criação de dificuldade adaptativa, resultando em uma melhor experiência do jogador. Silva, Silva e Chaimowicz (2015) desenvolveram um mecanismo de ajuste dinâmico da dificuldade por meio de inteligência artificial, demonstrando que a abordagem foi capaz de acompanhar as habilidades do jogador. A pesquisa de Robb e Zhang (2022) investigou o impacto do *flow* na experiência do jogador, utilizando um jogo de meteoros para comparar diferentes tipos de dificuldade. Os resultados indicaram que a dificuldade adaptativa foi a mais eficaz. O trabalho de Chanel et al. (2008) adotou uma abordagem baseada no reconhecimento de emoções dos jogadores para ajustar a dificuldade, mostrando que diferentes níveis de dificuldade podem gerar diferentes estados emocionais. Por fim, a pesquisa de Xue et al. (2017) desenvolveu um *framework* de dificuldade adaptativa que maximiza o engajamento do jogador ao otimizar as transições de estado.

Comparando esses estudos, podemos observar que a dificuldade adaptativa é uma área de pesquisa promissora, com abordagens variadas e resultados positivos em relação à experiência do jogador. O uso do *flow* e a consideração das emoções dos jogadores demonstram a importância de adaptar a dificuldade para atender às necessidades individuais e proporcionar uma experiência mais envolvente. A aplicação de inteligência artificial e a otimização global da dificuldade também mostram a capacidade de ajustar a jogabilidade de acordo com o progresso e as preferências dos jogadores.

No entanto, é importante destacar que cada abordagem tem suas próprias vantagens e desafios, e a escolha da melhor estratégia de dificuldade adaptativa depende do contexto e do tipo de jogo. A revisão de Zohaib (2018) fornece uma visão abrangente dessas abordagens e oferece diretrizes para a implementação da dificuldade adaptativa em diferentes gêneros de jogos.

Comparando esses estudos, podemos observar que a dificuldade adaptativa é uma área de pesquisa promissora, com abordagens variadas e resultados positivos em relação à experiência do jogador. O uso de teorias como o *flow* e a consideração das emoções dos jogadores demonstram a importância de adaptar a dificuldade para atender às necessidades individuais e proporcionar uma experiência mais envolvente. A aplicação de inteligência artificial e a otimização global da dificuldade também mostram a capacidade de ajustar a jogabilidade de acordo com o progresso e as preferências dos jogadores.

No entanto, é importante destacar que cada abordagem tem suas próprias vantagens e desafios, e a escolha da melhor estratégia de dificuldade adaptativa depende do contexto e do tipo de jogo. A revisão de Zohaib (2018) fornece uma visão abrangente dessas abordagens e oferece diretrizes para a implementação da dificuldade adaptativa em diferentes gêneros de jogos.

Com as pesquisas descritas anteriormente espera-se aprimorar a ferramenta SIMUPRO, desenvolvida neste trabalho, possibilitando a representação de diferentes gêneros de jogos com o intuito de liberar para o público o uso da ferramenta com bastante configurabilidade. Nos dois próximos capítulos são detalhados, respectivamente o processo de desenvolvimento e avaliação de cenários da ferramenta.

4 Desenvolvimento

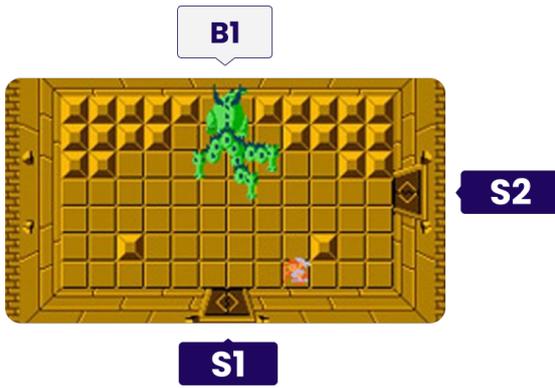
Esse capítulo detalha o desenvolvimento do SIMUPRO, um ambiente para planejamento de progressão de jogos, amparado por modelagem e subsequente simulação de agentes que representam o comportamento de diferentes tipos de jogadores. Entre a Seção 4.1 e a Seção 4.2, são discutidos os modelos de representação dos jogos no sistema. A Seção 4.3 e a Seção 4.4 trazem uma visão geral do ambiente de simulação e seus parâmetros. Por fim, na Seção 4.5 temos a estrutura do simulador. O projeto está disponível para acompanhamento no repositório do GitHub: <https://github.com/ufjf-gamelab/world-lab>.

4.1 Representação de desafios e objetivos de jogos em grafos

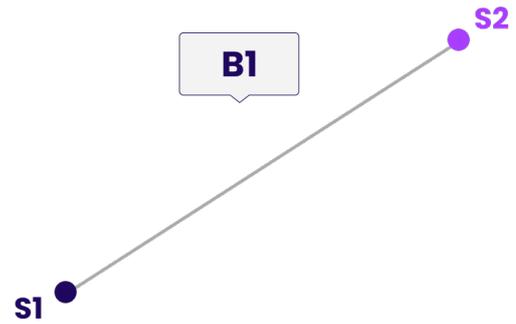
Os jogos apresentam os desafios em série para os jogadores, normalmente conhecidos como fases ou *levels*, em inglês. Essas fases, por sua vez, podem ser divididas em encontros com personagens não jogadores, desafios de habilidade, ou quebra-cabeças de lógica que apresentam a evolução da narrativa do jogo planejadas pelos *game designers*.

Esses desafios podem ser representados em grafos, por uma combinação de dois nós (por exemplo, rotulados por $S1$ e $S2$) ligados por uma aresta (por exemplo, com rótulo $B1$). O jogador (ou estado do jogo) pode ser representado estando em um desses nós, como $S1$, por exemplo. A conclusão, vitória ou solução do desafio é representada pela transição entre os nós via aresta, indo de $S1$ para $S2$ via $B1$. Cada nó e aresta possui suas próprias propriedades e características, como será apresentado na Seção 4.2. A construção da estrutura narrativa, ou de progressão narrativa do jogo, se dá através da interconexão de uma série de nós por arestas que representam uma parte do jogo, como ilustra a Figura 4.1.

Na Figura 4.1a, temos uma captura de tela de uma fase do jogo *The Legend of Zelda* (MIYAMOTO, 1986), no qual o inimigo verde é designado como $B1$, e os objetivos



(a) Captura de tela do jogo The Legend of Zelda (MIYAMOTO, 1986) onde os estados são representados pelas portas $S1$ (origem do personagem) e $S2$ (próximo objetivo) e o desafio pelo monstro verde. Fonte: do autor.



(b) Representação da Figura 4.1a em um grafo, onde o personagem precisa transitar entre as portas $S1$ e $S2$ tem que derrotar o desafio pela aresta $B1$. Fonte: do autor.

Figura 4.1: Representação de desafios de um jogo em grafos.

representados por $S1$ e $S2$. Observamos que o personagem está posicionado próximo à porta $S1$, que marca a entrada por onde ele veio, enquanto o objetivo $S2$ representa o local onde o personagem deseja chegar. No entanto, o objetivo $S2$ só pode ser alcançado quando o personagem derrota o inimigo $B1$. Essa representação neste trabalho pode ser visualizada na Figura 4.1b. A aresta $B1$ representa o desafio que corresponde ao monstro verde da Figura 4.1a, e o objetivo final é demarcado pelo nó $S2$ de cor lilás.

4.2 Estrutura de dados para os nós e arestas

Na implementação computacional, cada nó é representado por um objeto com um conjunto de propriedades em JavaScript. Algumas das propriedades são bem específicas da representação do jogo dentro do SIMUPRO, enquanto outras são para detalhes da representação dos grafos em si, necessárias para a biblioteca *CytoscapeJS*. Este conjunto pode ser observado no diagrama da Figura 4.2 e serão explicados a seguir.

A propriedade *data* contém as informações específicas utilizadas no SIMUPRO : o *id* é o identificador único do nó dentro do modelo; *label* o rótulo de referência para o designer; *churnCount* o número de vezes que um jogador desistiu no nó. As outras propriedades são específicas da biblioteca *CytoscapeJS* e são relacionadas ao estado de interação com o nó pelo usuário na interface do editor e sua posição e exibição.

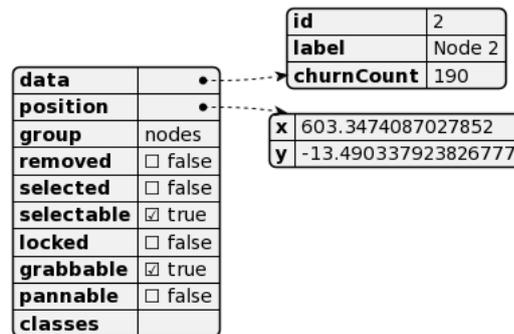


Figura 4.2: Representação da estrutura do nó onde as propriedades do SIMUPRO estão na direita com a posição do nó e as propriedades do *CytoscapeJS* na esquerda. Fonte: do autor.

A implementação da estrutura das arestas segue o mesmo formato da estrutura do nó e pode ser observada pelo diagrama da Figura 4.3.

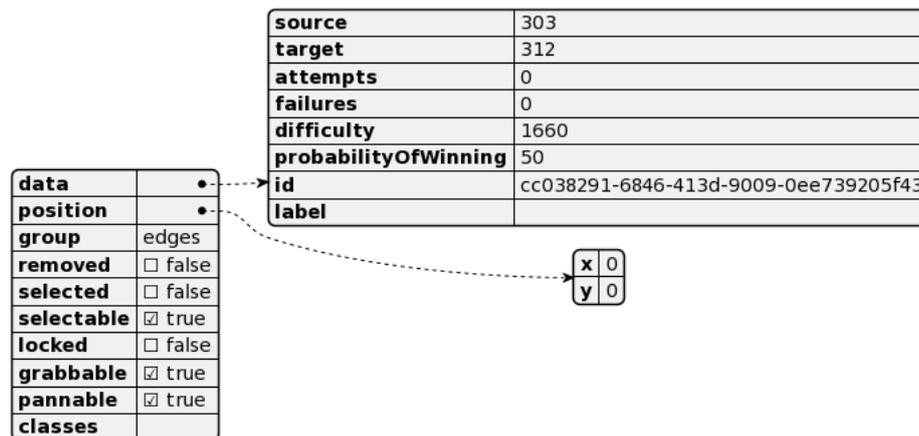


Figura 4.3: Representação da estrutura da aresta onde as propriedades do SIMUPRO estão à direita com a posição do nó e as propriedades do *CytoscapeJS* à esquerda. Fonte: do autor

As propriedades utilizadas no *Simupro* são armazenadas na propriedade *data*. A propriedade *data* da aresta, diferente do nó, contém informações sobre o nó de origem e de destino da aresta, o número de tentativas e falhas realizadas pelo jogador, probabilidade de vitória e a classificação do Elo da aresta.

4.3 Visão geral do simulador

O SIMUPRO é um ambiente de simulação criado para auxiliar os desenvolvedores de jogos, *game designers*, *level designers* ou qualquer outra função do usuário dentro da equipe a simular níveis e representar a progressão de diferentes modelos de jogadores. Com o

SIMUPRO é esperado que seja possível analisar como os níveis criados se comportam quando diversos tipos de jogadores, com comportamentos diferentes, o exploram. Dessa forma, espera-se obter dados para aprimorar a jogabilidade e a experiência dos jogadores ainda em tempo de *design*, sem substituir os testes de jogabilidade subsequentes. A Figura 4.4 ilustra o funcionamento do SIMUPRO.

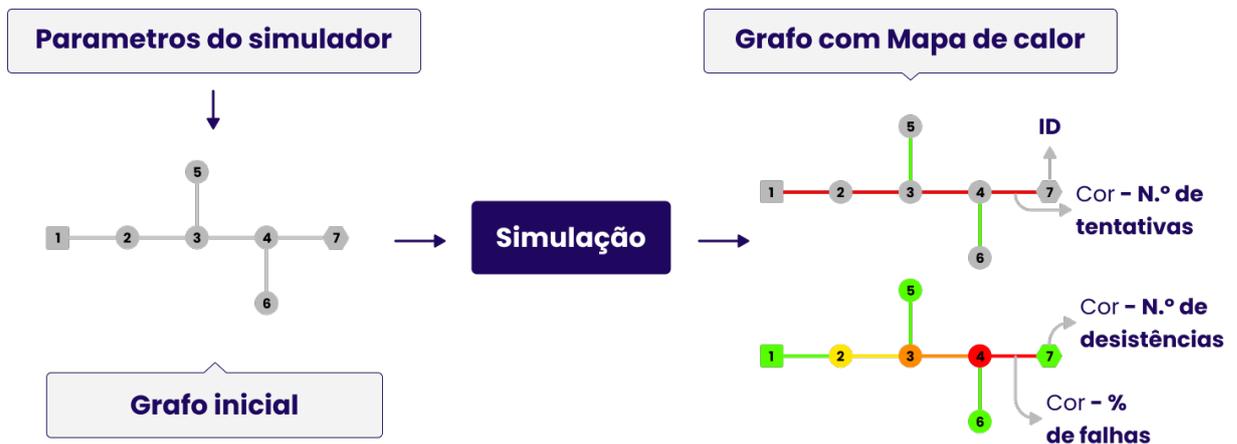


Figura 4.4: Ilustração do processo de utilização do SIMUPRO. O usuário preenche os parâmetros de simulação e junto com o grafo criado pelo mesmo gera os dados que podem ser observados pelos mapas de calor. As cores frias são usadas para representar valores pequenos e tons quentes para valores altos. Fonte: do autor

Para cada topologia, o usuário escolhe os parâmetros de simulação (detalhados na Seção 4.4) para definir um cenário de simulação e análise. A simulação consiste em criar um agente com o comportamento selecionado nos parâmetros e executar um grande conjunto de passeios pelo grafo pelo método de Monte Carlo, dada a representação estocástica dos desafios. Após a simulação, os dados obtidos podem ser observados por mapas de calor. Esses mapas são uma representação visual para identificar pontos de alta desistência ou grande número de falhas nos desafios. Cores frias (começando do verde) são usadas para representar valores pequenos e tons quentes para valores altos (vermelho). O mapa permite identificar a valores sobre as arestas pela sua espessura. Na Figura 4.4 temos alguns exemplos dos dados que o mapa de calor pode demonstrar. Com essa funcionalidade, o usuário pode explorar e interpretar as informações de forma mais clara e intuitiva.

4.4 Parâmetros da Simulação

Nesta seção, são apresentados os parâmetros utilizados nas simulações para criar cenários que representam diferentes ambientes e comportamentos dos jogadores. Os cenários de exemplo serão vistos posteriormente no Capítulo 5.

4.4.1 Início e fim da simulação

Para determinar o início e o fim da simulação, os parâmetros primeiro nó e último nó são utilizados. O usuário do SIMUPRO preenche os números dos identificadores de ambos os nós. Esses valores são utilizados nos algoritmos de navegação (Seção 2.8) determinados pelo tipo de jogador selecionado (Subseção 4.4.4). Na Figura 4.5, é possível visualizar a representação gráfica dos parâmetros mencionados: o nó inicial é identificado pelo número 10 e está destacado pelo formato de quadrado, enquanto o nó final é identificado pelo número 6 e está destacado pelo formato de hexágono.

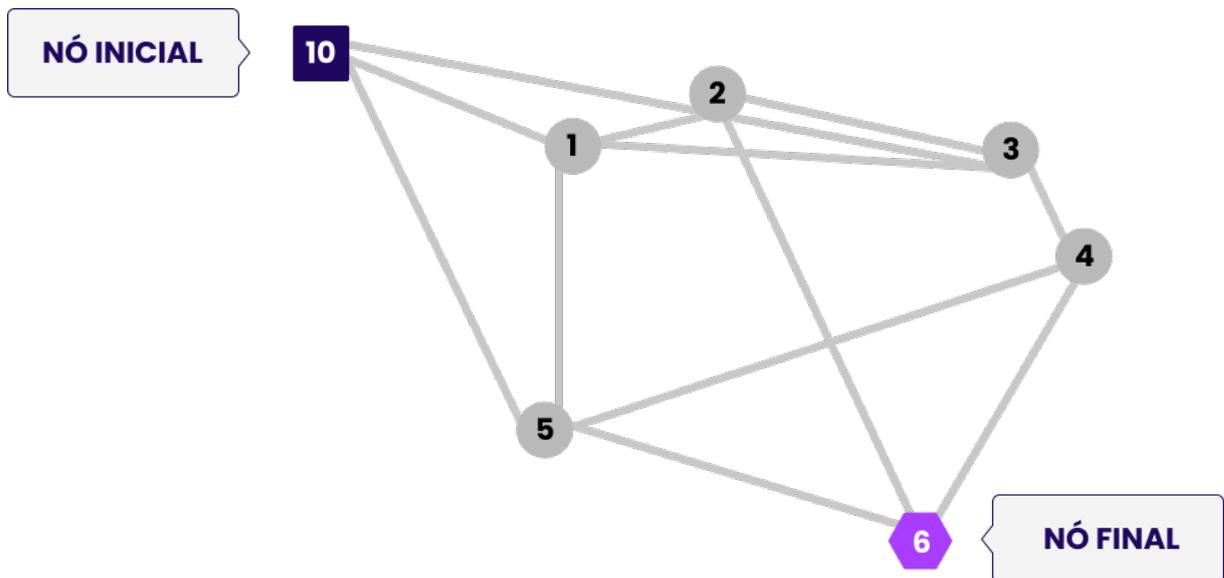


Figura 4.5: Representação gráfica do início e fim do jogo. O nó inicial é identificado pelo número 10 e é destacado com formato de um quadrado, simbolizando o ponto de partida do jogo. Já o nó final, representado pelo número 6, está destacado com o formato de um hexágono, indicando o objetivo a ser alcançado. Fonte: do autor.

4.4.2 Habilidade do agente

No sistema, é possível escolher duas formas de decidir se o agente obteve sucesso ou falha ao tentar um desafio: modelo por chance simples; e modelo por classificação de habilidade

baseado em Elo (visto na Seção 2.3). O projetista pode atribuir uma classificação inicial representando um jogador com alta ou baixa capacidade para concluir os desafios durante a simulação. Na atual versão do sistema, isso representa desde uma habilidade no controle dos mecanismos do jogo quanto um poder dado pela progressão do personagem, como equipamentos e atributos do jogo.

Essa classificação de habilidade também aparece como propriedade para uma “habilidade” do desafio, representado pela aresta (Seção 4.2). Este valor também é atribuído pelo projetista no momento que cria as arestas e também representa, de forma geral, a dificuldade do quebra-cabeça, o número e a quantidade de vida de inimigos.

4.4.3 Número de iterações

Este parâmetro determina o número de vezes que o simulador será executado com as mesmas configurações, permitindo ao usuário coletar dados estatísticos e avaliar a estabilidade e confiabilidade dos resultados obtidos. É importante destacar que o modelo não suporta a execução do simulador em diferentes perfis de jogadores ou dificuldades ao mesmo tempo. Em vez disso, ele trabalha apenas com um conjunto de configurações específicas selecionadas pelo usuário, como o tipo de jogador, o tipo de dificuldade, entre outros. Não é possível, por exemplo, realizar uma visão global atribuindo uma distribuição de habilidade de jogadores por uma curva normal, colocando os jogadores com habilidade muito alta ou muito baixa nos extremos.

4.4.4 Modelo de Perfil de Jogador

O parâmetro do modelo de perfil de jogador determina como o agente representando o jogador realiza o percurso do grafo, partindo do nó inicial até o nó final. Ele define o algoritmo de busca de acordo com um dos perfis discutidos na Seção 2.5. Para aplicação dos algoritmos de busca, foram usadas funções nativas do *CytoscapeJS*. Os perfis de socializadores e combatentes não serão modelados para a primeira versão deste trabalho por necessitarem de simular interações sociais e estéticas, que estão fora do nosso escopo no momento.

Perfil jogador explorador

Para representar o jogador de perfil explorador, a busca em profundidade é implementada através da função *cy.elements().dfs()*. Essa função recebe o nó inicial, o nó final e uma função heurística para definir o peso da aresta. O caminho recebido da busca é utilizado pelo agente para percorrer o grafo, com base na chance ou habilidade dos desafios.

Perfil jogador realizador

Como vimos anteriormente, o perfil de jogador realizador é aquele que se concentra em completar as missões principais do jogo, sem se preocupar muito com objetivos secundários ou exploração. Para representar esse tipo de jogador no grafo, foi utilizado o método *cy.elements().aStar()*. Esse método recebe o nó inicial, o nó final e uma função heurística para definir o peso da aresta. O caminho recebido da busca é utilizado pelo agente para percorrer o grafo da maneira mais eficiente possível, com base na chance ou habilidade dos desafios.

4.4.5 Modelos de desafios

O modelo de desafio determina se um jogador simulado, ou agente, consegue ou não superar um desafio durante a simulação. A cada iteração da simulação, um número é sorteado e se o agente superar a chance da aresta, ele faz a transição de um nó para o próximo nó através da aresta. Nesse trabalho, foram utilizados dois tipos de modelo de desafio: o modelo de desafio por aleatoriedade ou chance fixa; e o modelo de desafio por habilidade.

Modelo de desafio por chance fixa

No modelo de desafio por aleatoriedade, a dificuldade é determinada pela propriedade da aresta chamada de “probabilidade de vitória”, conforme explicado na Seção 4.2. Durante o duelo, um número aleatório é gerado para o agente, variando de 0.0 a 100.0. Se esse número for maior do que a propriedade da aresta, ocorre a vitória. Por exemplo, se uma aresta tem 40% de chance de vitória para o jogador, o jogador precisa obter um número igual ou maior que 60% (100 - 40) para vencer o duelo.

Modelo de desafio por habilidade (*rating*)

Para o modelo baseado em habilidade, a dificuldade é calculada pela diferença entre a habilidade do jogador (seja no controle do jogo em conjunto com uma quantidade de itens ou estatísticas que contribuem para o personagem dele ter sucesso); e da habilidade do desafio (que representa de forma abstrata a complexidade do quebra-cabeça, quantidade de vida e dano de um monstro ou número de inimigos). Foi adotado o cálculo da probabilidade de sucesso utilizando a estimativa de sucesso do sistema de classificação Elo. A fórmula do Elo dada pela Equação 4.1 é utilizada para calcular a probabilidade de vitória do agente com base na diferença de suas habilidades contra a habilidade do desafio, conforme visto na Seção 2.3.

Na fórmula do Elo, temos a habilidade do agente representada por R_1 , enquanto a dificuldade da tarefa é representada por R_2 . Por exemplo, suponha que um agente tenha uma habilidade de $R_1 = 1800$, e uma aresta específica tenha uma dificuldade de $R_2 = 1600$. Substituindo esses valores na fórmula do Elo, obtemos como resultado 0.62. Isso significa que o agente possui uma probabilidade de vitória de aproximadamente 62% contra essa aresta específica.

$$P = \frac{1}{1 + 10^{\frac{R_2 - R_1}{400}}} \quad (4.1)$$

onde P é a probabilidade de vitória do agente com *rating* R_1 sobre o inimigo com *rating* R_2 .

4.4.6 Modelo de desistência (*churn*)

O modelo de desistência determina o número de falhas que um jogador precisa ter para desistir do jogo. Esse modelo é baseado no trabalho de Xue et al. (2017), visto na Seção 3.7, que utilizou um estado para desistência dos jogadores em sua pesquisa. Dois tipos de modelos foram definidos: desistência por limite de tentativas e desistência por perda de *Flow*.

Modelo de desistência por limite de tentativas

O modelo de desistência por limite de tentativas é usado para determinar um ponto em que o jogador decide parar de jogar após um número pré-determinado de tentativas sem sucesso contra um desafio. Esse modelo reconhece que os jogadores podem encontrar desafios que consideram muito difíceis ou frustrantes, e com isso desistem. Nesse modelo, é estabelecido um número específico de tentativas permitidas até que o agente desista de jogar. O modelo pode ser configurado para interromper o jogo após 1, 2 ou 3 derrotas seguidas contra um mesmo desafio.

Modelo de desistência por *flow*

O modelo de desistência por perda de *flow* é baseado na teoria do *flow* proposta por Csikszentmihalyi (1990). O jogador abandona o jogo (*churn*) quando perde o estado de *flow*, ou por frustração (sua habilidade é muito maior que os desafios) ou ansiedade (sua habilidade é muito menor que os desafios). Em ambos os casos, há a perda de motivação para continuar jogando e ele interrompe sua progressão no jogo.

A aplicação desse modelo se baseia no trabalho de Silva, Silva e Chaimowicz (2015), conforme apresentado na Seção 3.3. A zona de *flow* é representada na escala pelos valores entre 40 e 60. Valores acima de 60 indicam uma zona de ansiedade, em que o percentual de vitórias do jogador é baixo, enquanto valores abaixo de 40 indicam uma zona de tédio, onde o jogador está vencendo a grande maioria das partidas. Se o jogador permanecer mais de 3 duelos fora da zona de *flow*, ele desiste do jogo.

Para esse modelo, a escala de estresse e tédio são continuamente atualizados ao longo da simulação. Isso significa que, se o agente enfrentar muitos inimigos fáceis no começo da simulação, ele pode desistir antes de enfrentar inimigos mais difíceis no final do mapa devido ao tédio acumulado. Essa dinâmica indica que a progressão adequada do desafio é importante para manter o interesse do agente ao longo da experiência de jogo.

4.4.7 Modelo de progressão do jogador

O modelo de progressão do jogador determina se ele melhora a capacidade de superar os desafios do jogo, ou pelo aumento do poder do personagem, ou pelo desenvolvimento de

sua habilidade e maestria dos elementos do jogo. Neste trabalho, foram abordados dois modelos distintos: o primeiro é o modelo sem progressão ou progressão nula, no qual a habilidade do jogador permanece inalterada ao longo do jogo; o segundo é o modelo de progressão linear, no qual o jogador aprimora suas habilidades gradualmente à medida que fica mais tempo no jogo. Um jogo bem-planejado pode manter o jogador no *flow* se seguir essa curva.

Modelo sem progressão

O modelo sem progressão é utilizado para observar ou isolar o efeito da habilidade do jogador em um jogo. Ele é aplicado quando se deseja analisar o impacto da habilidade individual independentemente de qualquer progressão ao longo do tempo. Esse modelo foi feito para simular situações em que um jogador não investe muita atenção no jogo ou joga apenas algumas poucas partidas antes de apresentar um efeito significativo em seu desempenho. A Equação 4.2 descreve a implementação da progressão nula neste trabalho, em que a habilidade do jogador ao longo do tempo $S(t)$ é a mesma do início de seu contato com o jogo S_0 . Esta permanece constante até o jogador completar o jogo ou desistir de jogá-lo (*churn*).

$$S(t) = S_0 \tag{4.2}$$

Modelo de progressão linear

O modelo de progressão linear foi desenvolvido para representar jogadores que melhoram suas habilidades ao longo do tempo. Nesse modelo, a habilidade do jogador durante a partida ($S(t)$) varia de forma proporcional à quantidade de partidas jogadas, conforme a equação apresentada na Equação 4.3. Nessa equação, S_0 representa a habilidade inicial do jogador, r_g é a taxa de crescimento da habilidade, que indica o aumento da habilidade por vitórias do agente, e k é o número de vitórias acumuladas pelo jogador.

$$S(t) = S_0 + r_g \cdot k \tag{4.3}$$

4.4.8 Modelo de dificuldade

O modelo de dificuldade determina a classificação da dificuldade, ou seja, como a dificuldade dos desafios muda ao longo do jogo. Para esse trabalho foram definidas dois tipos de modelos: não adaptativo e adaptativo.

Modelo de dificuldade não adaptativa

O modelo não adaptativo implica que o desafio do jogo permanecerá constante ao longo da partida, ou seja, os valores das arestas permanecerão com os mesmos valores ao decorrer do jogo. Porém, as dificuldades podem ser ajustadas manualmente pelo projetista.

$$R'_{pe} = R_{epo} + k(S_p - E_p) \quad (4.4a)$$

Modelo de dificuldade adaptativo

Este modelo realiza ajustes com base nas informações coletadas após a simulação. Primeiramente, é identificado o nó com a maior taxa de desistência, juntamente com a aresta que apresenta o maior percentual de falhas em relação ao número de tentativas. Em seguida, são selecionadas as duas arestas adjacentes a esse desafio específico. Utilizando essas informações, é calculada uma média dos valores das arestas selecionadas junto com a dificuldade media do grafo todo, a fim de determinar a nova dificuldade do desafio.

Os ajustes feitos pelo modelo também são influenciados pelos modelos de desafio e desistência selecionados. Se o modelo de desafio baseado em habilidade for escolhido na simulação, um *rating* é atribuído ao agente, que representa sua habilidade como jogador. Esse *rating* é utilizado para ajustar as dificuldades dos desafios presentes no jogo. O processo de atualização do *rating* aparente envolve o desempenho real do agente nos desafios (se ele teve sucesso ou não).

Com base nesse *rating* e na média das dificuldades das arestas vizinhas, as dificuldades dos desafios são ajustadas, tornando o jogo mais desafiador ou mais acessível, dependendo do desempenho do agente. Esse ajuste é realizado usando a equação Equação 4.4a, na qual o novo *rating* do agente (R_{pe}) é atualizado com base no *rating* anterior do agente (R_{epo}), o fator de escala K , o resultado esperado (E_p) e o resultado observado (S_p). O *ra-*

ting anterior representa a estimativa de habilidade do jogador antes da partida, enquanto o fator de escala K determina a magnitude do ajuste e possui um valor constante de 32 no modelo de Elo padrão. O resultado esperado é a probabilidade estimada de vitória do jogador, calculada com base nas diferenças de habilidade entre os jogadores, conforme explicado na seção relacionada ao modelo de desafio (Subseção 4.4.5).

Se o modelo de desistência baseado em *flow* (Seção 4.4.6) é escolhido, a escala do modelo de *flow* é utilizada para determinar como a dificuldade deve ser ajustada. Pois a desistência pode ocorrer devido ao tédio ou à ansiedade, diferentemente do modelo baseado em tentativas (Seção 4.4.6), que considera apenas a quantidade de vezes que o agente tenta vencer o desafio.

4.5 Estrutura e funcionalidades do SIMUPRO

O SIMUPRO é composto por três partes principais: o painel de Configuração, o editor de Topologia e as ferramentas de edição e visualização.

Na Figura 4.6, você pode visualizar uma captura de tela do SIMUPRO, onde os três componentes estão destacados por letras. O painel de configuração é identificado pela letra "A", o editor de topologia pela letra "B" e, por último, as ferramentas de edição e visualização são representadas pela letra "C".

Painel de configuração

No configurador de simulação, os usuários têm a oportunidade de escolher os parâmetros de simulação (Figura 4.6). Esses parâmetros são responsáveis por determinar o cenário que será simulado. Por meio de diferentes combinações desses parâmetros, é possível criar uma variedade de cenários (conforme discutido no Capítulo de Cenários Capítulo 5) e observar os efeitos resultantes nos dados gerados a partir das simulações de centenas de jogadores.

Além disso, o componente do configurador de simulação também oferece uma funcionalidade importante: o carregamento e salvamento do modelo de jogo. Essa funcionalidade permite que os usuários criem e compartilhem diferentes configurações de simulação.

Editor de topologia

No editor de topologia temos a representação do progresso na narrativa ou *level design* a ser desenvolvido pelo projetista. A topologia é criada pela posição dos nós e suas conexões por arestas. Nela o autor pode criar e visualizar a estrutura do jogo. Ela também funciona como saída para a posterior análise dos dados obtidos durante a simulação mediante diferentes visualizações dos dados selecionada no componente de ferramentas de edição e visualização.

Ferramentas de edição e visualização

Essas ferramentas oferecem uma interface na qual os usuários podem habilitar a visualização dos dados da simulação no grafo, que vão aparecer diretamente sobre os nós e arestas. Um mapa de calor é exibido mostrando dados da simulação. Dados como a taxa de falhas e/ou tentativas em cada aresta, o número de desistências em cada nó, a probabilidade de vitória de cada aresta e a taxa de sucesso em cada aresta podem ser vistos. Para as arestas, esses dados podem ser vistos pela cor, texto em cima das mesmas ou pela largura. Além disso, as funções de edição e manipulação do grafo permitem fazer ajustes nas propriedades dos nós e das arestas.

Por fim, a ferramenta de edição e visualização oferece uma interface que permite aos usuários explorar e analisar os dados da simulação no grafo. Os nós e arestas exibem diretamente as informações relevantes, proporcionando uma representação visual dos dados.

No próximo capítulo, alguns modelos de jogos são apresentados na forma de cenários para avaliar a ferramenta e alguns comportamentos dos modelos.

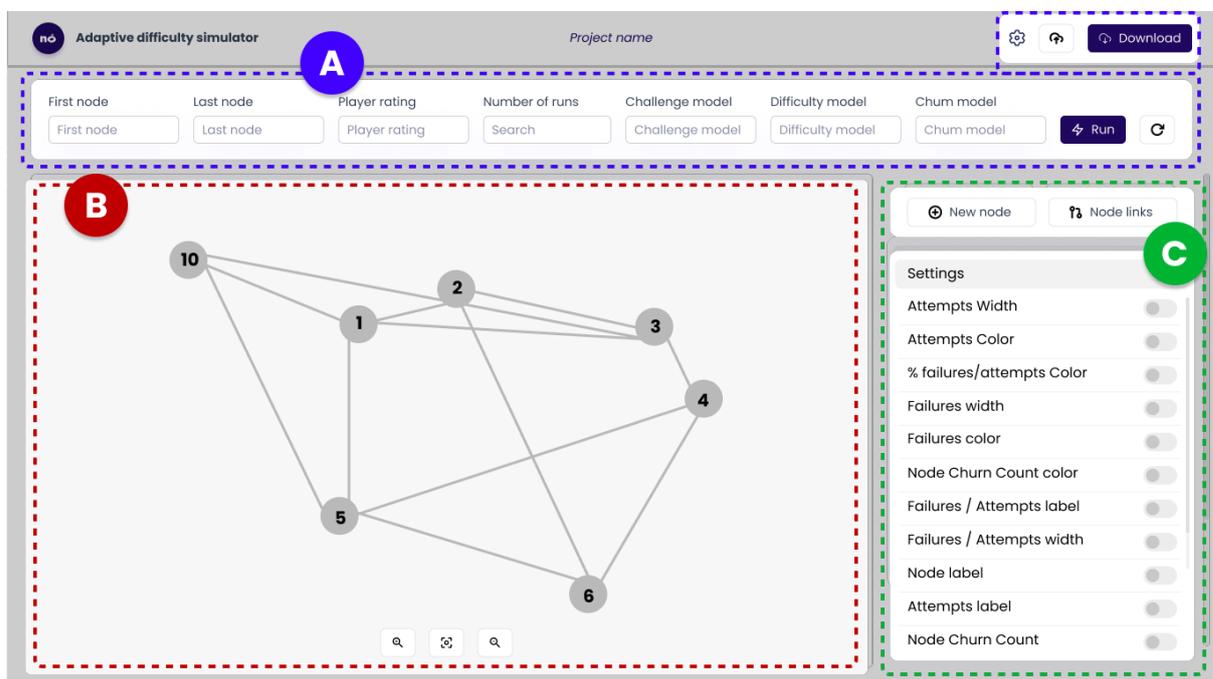


Figura 4.6: Captura de tela do SIMUPRO. Temos os três componentes separados e designados por letras. Temos o painel de configuração designado pela letra “A”, o editor de topologia pela letra “B” e por último, as ferramentas de edição e visualização pela letra “C”. Fonte: do autor.

5 Avaliação de Cenários

Este capítulo discute um conjunto de diferentes cenários no SIMUPRO. Cada cenário representa uma progressão de narrativa e dificuldade criada pelo projetista bem como uma combinação dos parâmetros de simulação vistos.

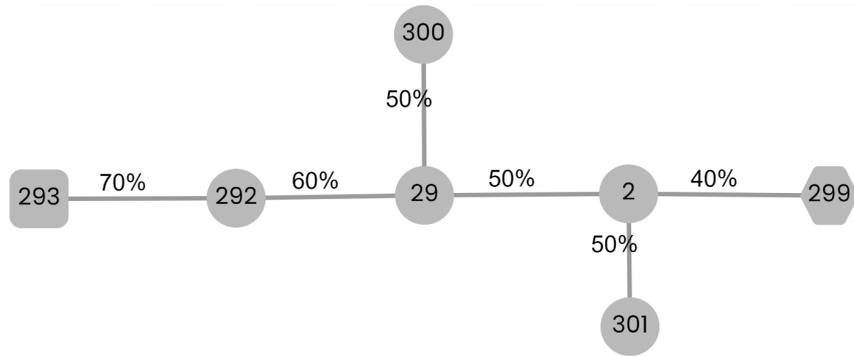
5.1 Cenário de referência

Este cenário tem como objetivo mostrar como um jogo com narrativa linear pode ser representado no SIMUPRO. Exemplos de jogos com estrutura semelhante incluem Super Mario Bros. Deluxe (NINTENDO, 2019) e Rayman Legends (ANCEL; HÉRAL; MONTPELLIER, 2013). A execução da simulação consiste em 1000 agentes tentando passar pelos desafios criados.

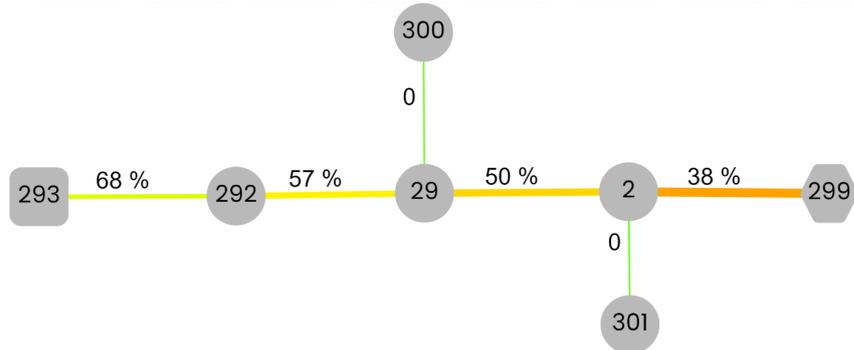
Este cenário não inclui parâmetros como dificuldade adaptativa, modelo de progressão, habilidade do jogador e modelo de desistência, que serão abordados nos próximos cenários, juntamente com os ajustes manuais e ajustes pela dificuldade adaptativa. Na Figura 5.1a, temos a representação gráfica deste cenário, onde cada aresta mostra a probabilidade estimada de vitória do agente, definida pelo autor do mapa.

Observa-se a chance de vitória decrescente nas arestas, planejada pelo projetista, para que o jogador encontre cada vez mais dificuldade ao avançar pelo caminho. O primeiro desafio possui uma probabilidade de vitória de 70%, que diminui em decrementos de 10% até chegar ao último desafio, com uma probabilidade de vitória de 40%.

A Figura 5.1b apresenta o resultado da simulação de 1000 agentes percorrendo o mapa, na forma de um mapa de calor. Para representar os dados, foram selecionadas as seguintes configurações: o percentual de derrotas em relação às tentativas e o percentual de sucesso nos duelos. As cores e larguras das arestas indicam o primeiro, enquanto os números nas arestas representam o sucesso nos duelos. Como esperado, a Figura 5.1b, observa-se que o número de vitórias dos agentes diminui ao longo do grafo, seguindo a tendência esperada com base na probabilidade estimada de vitória apresentada na Fi-



(a) Modelo de jogo criado pelo projetista, as chances são definidas no momento da criação.



(b) Mapa de calor após 1000 agentes com perfil realizador percorrerem o mapa. O resultado está de acordo com valor teórico esperado do cenário. Temos a probabilidade de vitória dos agentes sendo representada pela cor e o número nas arestas.

Figura 5.1: Modelagem e resultado da simulação do cenário de referência. Fonte: do autor.

gura 5.1a com uma diferença de 3% em relação às expectativas.

Além disso, nota-se que o agente da simulação não buscou completar os objetivos secundários representados pelos nós 300 e 301. Essa observação está alinhada com o perfil realizador escolhido para os agentes simulados, que têm como objetivo principal concluir o jogo o mais rápido possível, buscando o caminho mais direto para alcançar o nó final.

As próximas seções apresentam cenários que são variações deste apresentado como referência. Neles é demonstrado como a mudança de um parâmetro influencia na simulação.

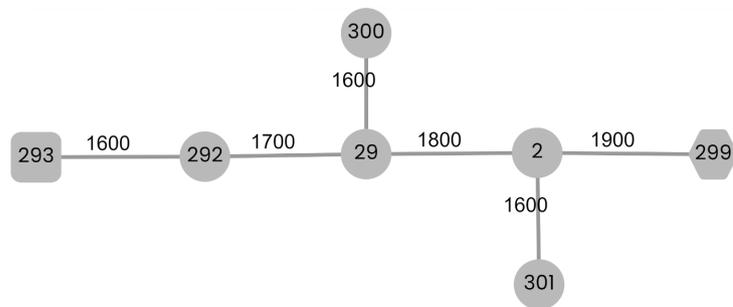
5.2 Cenário 1: Modelo de desafio por habilidade

Neste cenário, são utilizados a mesma topologia e parâmetros do cenário de referência. No entanto, é feita uma alteração no modelo de desafio, substituindo o modelo de desafio

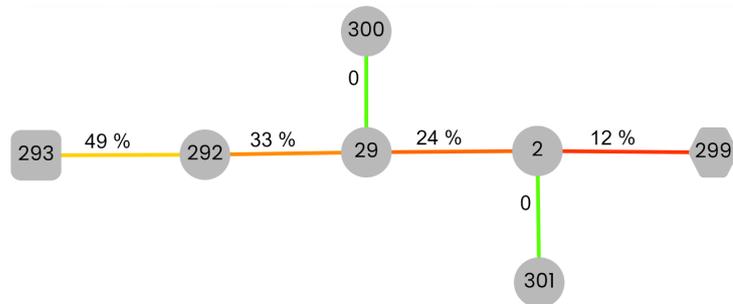
por chance fixa pelo modelo de desafio por habilidade. Arbitrou-se a habilidade média para os jogadores simulados como 1600 e também o desafio inicial. Então, aplicou um crescimento nos desafios em incrementos de 100, como pode ser visto na Figura 5.2a. Dessa forma, a probabilidade de vitória do jogador é calculada com base na habilidade do mesmo em relação à habilidade do desafio, conforme visto na Seção 4.4.5.

Após a simulação, a Figura 5.2b apresenta o mapa de calor que mostra o percentual de vitórias dos agentes por meio das cores, larguras e números nas arestas. É possível observar que o número de vitórias diminui ao longo do percurso, o que era esperado, devido ao aumento da diferença entre os *Elo rating* das arestas, junto com a falta de um modelo de progressão de habilidade para o jogador.

Para verificar o modelo, é possível calcular a probabilidade de vitória esperada usando a habilidade de um jogador com *Elo rating* de 1600. Por exemplo, na segunda aresta entre os nós 292 e 29, a probabilidade de vitória é de 33%. Comparando com a probabilidade esperada de vitória de um jogador de 1600 enfrentando um oponente de 1700, que é de aproximadamente 36.05%, podemos observar novamente uma variação de apenas 3%, que diminui se o número de iterações aumenta.



(a) Modelo de jogo criado com modelo de desafio por habilidade.



(b) Resultado da simulação do modelo de desafio por habilidade.

Figura 5.2: Cenário 1: modelo de desafio por habilidade usando o *Elo rating*. Fonte: do autor.

5.3 Cenário 2: Modelo de progressão de habilidade

No Cenário 1 (Seção 5.2), foi observada uma queda acentuada no número de vitórias dos agentes ao utilizar o modelo de desafio por habilidade. Isso se deve ao aumento constante e linear da dificuldade nos desafios. À medida que o jogador com habilidade constante avança, os desafios se tornam cada vez mais difíceis para ele. Ao habilitar o modelo de progressão linear de habilidade do jogador, o efeito de maestria ou recompensas que aumentam o poder do personagem pode ser representado.

No modelo de progressão linear, a habilidade do jogador simulado aumenta em 100 pontos após cada desafio vencido, seguindo a escala de Elo. A Figura 5.3 apresenta uma nova simulação com 10.000 iterações sobre o mesmo grafo e parâmetros do Cenário 1 (Figura 5.2b) utilizando o modelo de progressão de personagem. O aumento no número de iterações serve para eliminar anomalias matemáticas causadas pela quantidade baixa de duelos. Utilizando a escala de Elo podemos observar que o percentual de vitória dos agentes se mantém constante com uma variação de apenas 2% do resultado esperado, em contraste com os resultados do Cenário 1 (Figura 5.2b), onde o índice de vitória diminui rapidamente, começando em 49% e terminando em 12%. Esse resultado está conforme o esperado, uma vez que o nível de habilidade do jogador está se alinhando com a dificuldade dos desafios, evitando uma disparidade significativa entre a habilidade de ambos.

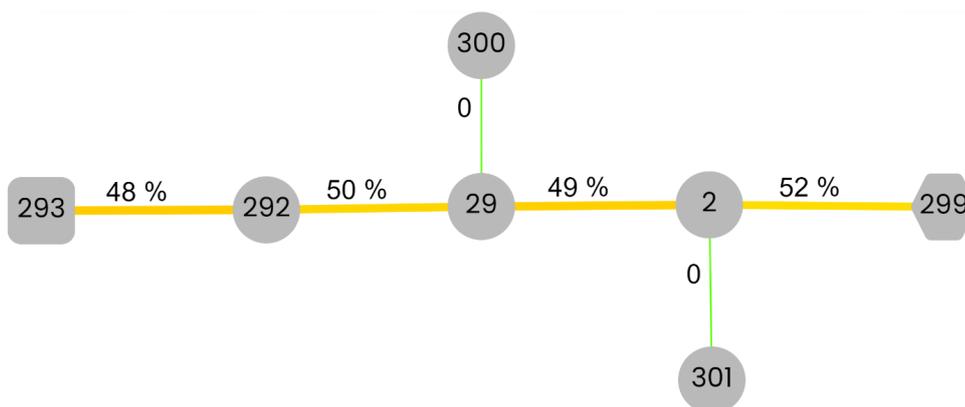


Figura 5.3: Mapa de calor depois de 10.000 iterações simuladas do Cenário 2 habilitando o modelo de progressão de jogador. Fonte: do autor.

5.4 Cenário 3: Modelo de perfil de jogador explorador

Neste cenário, é explorado como o perfil do jogador explorador influencia o percurso dos agentes pelo grafo. Como este perfil vai tentar percorrer todos os nós, ao mudarmos o perfil do jogador, esperamos que a probabilidade de vitória nos desafios permaneça próxima à projetada e que também os agentes tentem vencer todas as arestas do mapa.

Na Figura 5.4, temos o resultado da simulação com 10.000 agentes utilizando os mesmos parâmetros do cenário de referência, mas com o perfil explorador. Através das cores e números nas arestas, podemos observar que os agentes obtiveram 49% de vitória nos desafios que anteriormente não foram confrontados, o que indica que a busca em profundidade foi bem-sucedida. Além disso, os dados observados estão bem próximos do percentual de vitória teórico, como mostrado na Figura 5.1a, com uma variação de no máximo 6% pela teoria do cálculo de erro.

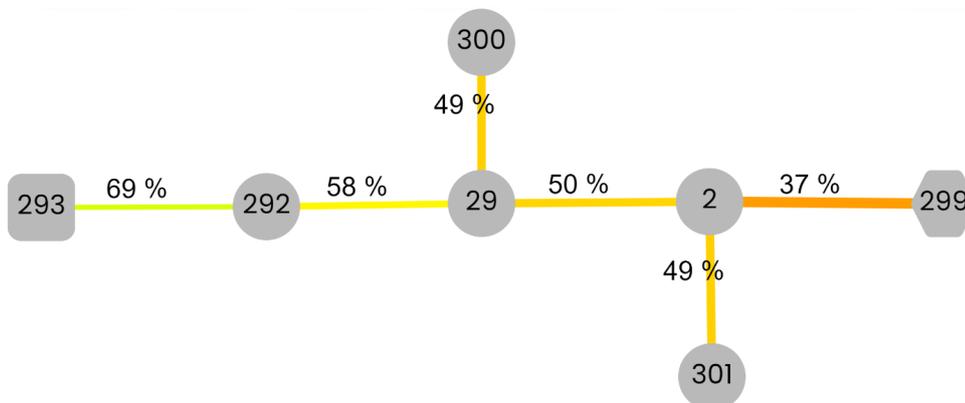


Figura 5.4: Mapa de calor para o Cenário 3 com perfil de jogador explorador que realiza uma busca em profundidade. Fonte: do autor.

5.5 Cenário 4: Modelo de desistência por *flow* (tédio)

Neste cenário, temos o funcionamento do modelo de desistência baseado no *flow* (descrito na Seção 4.4.6), que pode levar à desistência por tédio ou ansiedade nos jogadores simulados. Mais especificamente, neste cenário, estamos considerando a desistência por tédio, que ocorre quando os desafios se tornam muito fáceis para o jogador simulado, levando-o a desistir antes de completar o jogo. Nos cenários anteriores, utilizamos o mo-

delo de desistência por tentativa, que permitia um número máximo de tentativas antes da desistência ocorrer.

Na Figura 5.5a, temos a representação gráfica deste cenário, que utiliza os mesmos parâmetros do cenário de referência, mas com a probabilidade de vitória em cada desafio definida como 90%, simulando um jogo com desafios muito fáceis.

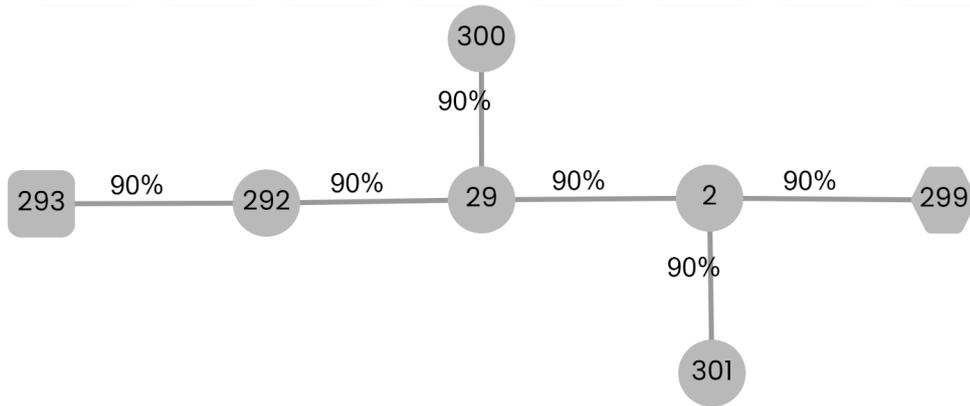
A Figura 5.5b apresenta o resultado obtido neste cenário após uma simulação de 10.000 agentes. Nessa figura, a cor e a largura das arestas representam o número de tentativas realizadas, enquanto os números nas arestas indicam o percentual de vitória. Além disso, as cores dos nós representam a quantidade de desistências por tédio.

Ao analisarmos a figura, podemos observar que o percentual de vitória se mantém o mesmo ao longo do grafo, porém o número de tentativas diminui. Isso é evidenciado pela variação na cor e espessura das arestas. Conseqüentemente, observamos um aumento no número de desistências nos nós 28 e 2. Esses resultados estão conforme o esperado, pois à medida que o jogador simulado enfrenta desafios muito fáceis e vence facilmente, a escala de fluxo inclina para o tédio. Se essa facilidade persistir, é esperado que o jogador desista de jogar.

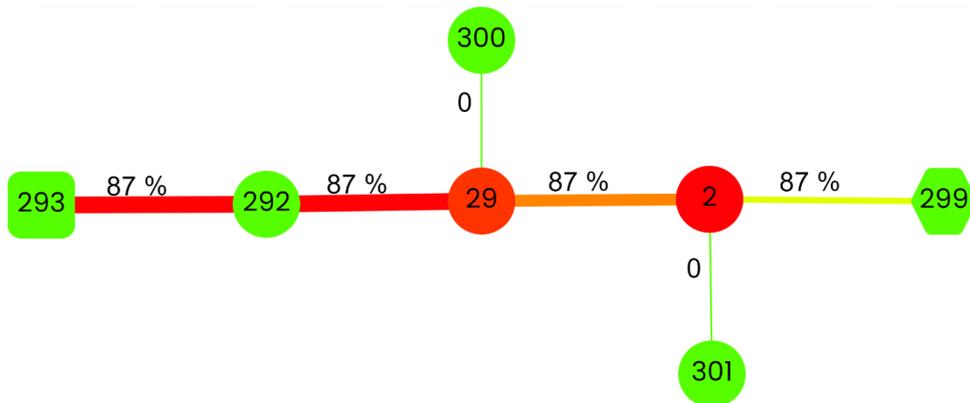
5.6 Cenário 5: Modelo de desistência por *flow* (ansiedade)

Para representar a desistência por ansiedade, neste cenário temos um ambiente altamente desafiador, com baixa chance de sucesso. Uma variação do Cenário 4 foi utilizada, mantendo a mesma topologia e parâmetros, mas ajustando a probabilidade de vitória esperada dos agentes. Essa probabilidade começa em 40% e diminui gradualmente em decrementos de 10% até alcançar o valor mínimo de 10%. Dessa forma, podemos simular a experiência de jogadores que enfrentam dificuldades significativas, levando à possibilidade de desistência motivada pela ansiedade no modelo de *flow*.

Na Figura 5.6a, temos o grafo representando o cenário, com as probabilidades estimadas de vitória nas arestas, conforme definido pelo modelo de desafio por chance fixa (Seção 4.4.5). Em seguida, foi realizada uma simulação de 10.000 agentes e na



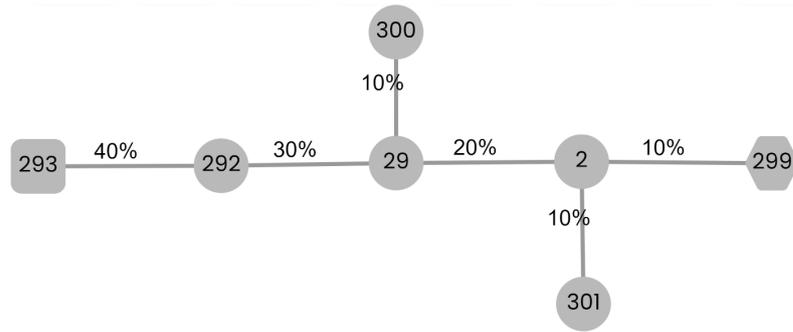
(a) Grafo utilizado no cenário 4 com a probabilidade de vitória esperada do agente nas arestas pelo modelo de chance.



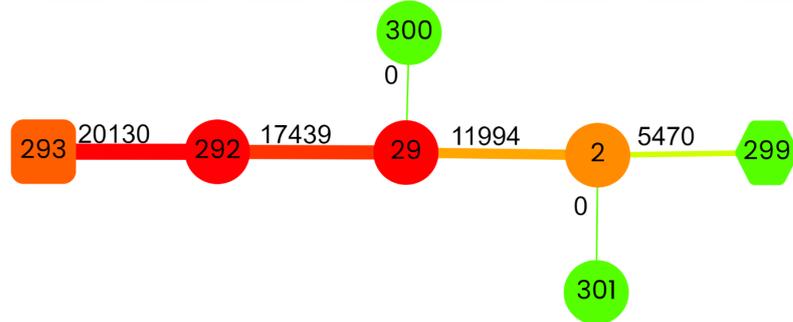
(b) Mapa de calor do Cenário 4 que apresenta a desistência dos jogadores pela cor do nó. Temos também a quantidade de tentativas representada pela espessura da aresta junto com a probabilidade de vitória dos agentes representada pela cor e pelo número nas arestas. Fonte: do autor

Figura 5.5: Representação gráfica do Cenário 4 com modelo de desistência (*churn*) por tédio.

Figura 5.6b é possível ver os resultados através do mapa de calor. A quantidade de tentativas é representada pelo número, cor e largura das arestas, enquanto a cor dos nós indica o número de desistências por ansiedade. É possível observar que houve uma redução significativa da quantidade de tentativas ao longo do grafo, devido ao aumento das desistências motivadas pela dificuldade. Isso pode ser visto no desafio final, representado pela aresta entre os nós 2 e 299, onde restaram apenas 2.286 jogadores, dos 10.000 iniciais. Esses resultados estão alinhados com as expectativas do estudo.



(a) Cenário 5 com a probabilidade de vitória esperada decrescente.



(b) Mapa de calor após 10.000 agentes da simulação do Cenário 5: com a quantidade de tentativas são exibidas nas arestas e a espessura indica a quantidade de agentes que conseguiram passar pelo desafio.

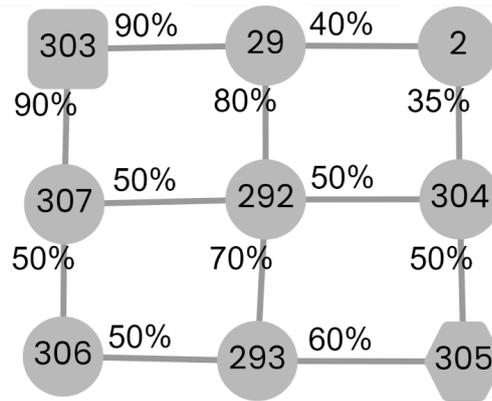
Figura 5.6: Representação gráfica do Cenário 5 com os resultados representados por um mapa de calor. Fonte: do autor.

5.7 Cenário 6: Topologia em grade

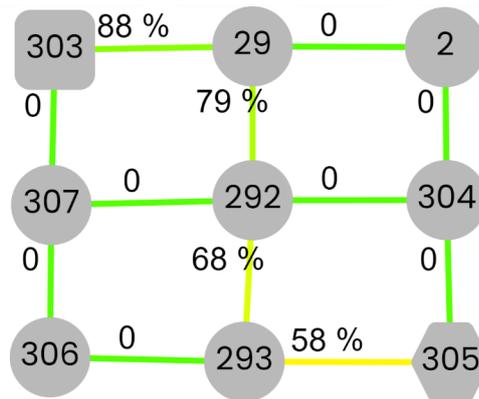
Para esse cenário é demonstrado como os jogos com topologia em grade são representados no SIMUPRO. Os mesmos parâmetros do cenário de referência são utilizados, ou seja, o modelo de desafio por chance e o perfil do jogador realizador. A mudança do cenário ocorre no formato do grafo, juntamente com os nós iniciais e finais.

A Figura 5.7a ilustra a representação gráfica deste cenário com a probabilidade de vitória nas arestas. Também é possível ver múltiplos caminhos possíveis até o fim do jogo. Tendo isso em mente, é esperado que a busca A* do modelo de perfil selecionado selecione o caminho mais fácil.

Após realizar uma simulação de 10.000 agentes, os resultados obtidos foram apresentados no mapa de calor na Figura 5.7b. Ao observar a imagem, é possível constatar que o percentual de vitória está conforme o esperado, variando no máximo 2%. Além disso, nota-se que o algoritmo de busca A* selecionou o caminho mais fácil, ou seja, aquele com a maior probabilidade de vitórias.



(a) Probabilidade de vitória esperada do agente nas arestas pelo modelo de chance.



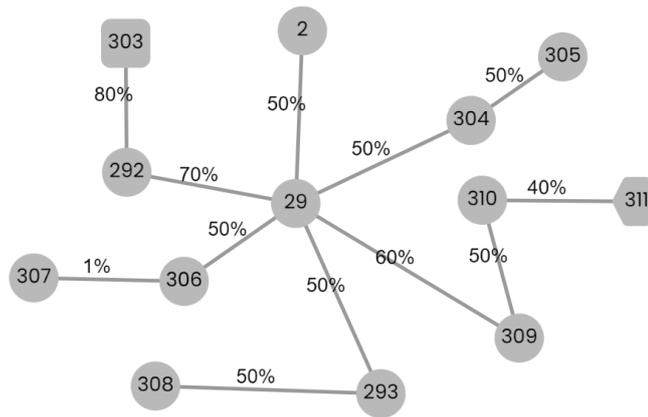
(b) Mapa de calor após 10.000 agentes. Os resultados obtidos estão de acordo com valor teórico esperado do cenário.

Figura 5.7: Representação gráfica do Cenário 6 com modelo de chance e perfil realizador. Fonte: do autor.

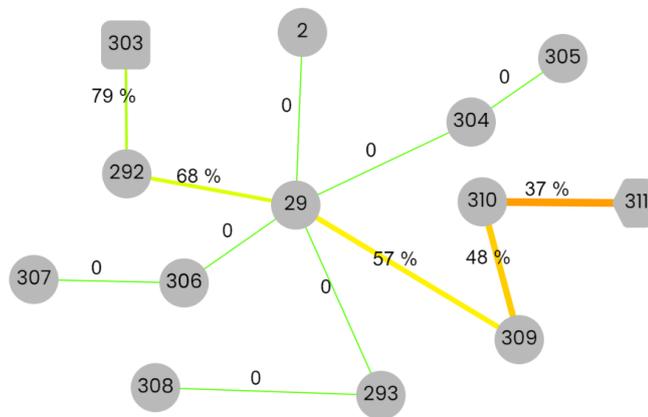
5.8 Cenário 7: Topologia de rede

Os mesmos parâmetros do cenário de referência são utilizados, ou seja, o modelo de desafio por chance e o perfil do jogador realizador. A mudança do cenário ocorre no formato do grafo, juntamente com os nós iniciais e finais, que foram alterados para 303 e 311, respectivamente. A Figura 5.8a ilustra a representação gráfica deste cenário com as probabilidades de vitória nas arestas.

Após realizar uma simulação de 10.000 agentes, os resultados obtidos foram apresentados no mapa de calor na Figura 5.8b. Ao observar a imagem, é possível constatar que o percentual de vitória está conforme o esperado, variando no máximo 3%.



(a) Cenário 7 com a probabilidade de vitória esperada do agente nas arestas pelo modelo de chance.



(b) Mapa de calor após 10.000 iterações da simulação do Cenário 7. Os resultados obtidos estão de acordo com valor teórico esperado do cenário.

Figura 5.8: Representação gráfica do Cenário 7 com os resultados representados por um mapa de calor. Fonte: do autor.

5.9 Cenário 8: Aplicação de um ajuste manual

Este cenário tem como objetivo demonstrar os ajustes manuais que podem ser realizados nos elementos do grafo e como essas alterações afetam a dinâmica de uma simulação.

No contexto deste cenário específico, temos a representação de um jogo de topologia linear onde foram utilizados os seguintes parâmetros de simulação: o nó inicial foi definido como 293, o nó final como 299 e o número total de iterações foi estabelecido em 1000. O modelo de desafio de chance fixa foi escolhido, juntamente com o perfil do jogador explorador. Além disso, foi utilizado o modelo de desistência com limite de 2 tentativas. Os modelos de progressão e dificuldade adaptativa não foram aplicados neste cenário.

Na Representação gráfica do Cenário (Figura 5.9a), podemos observar a pro-

habilidade de vitória esperada do agente em cada aresta, juntamente com a identificação dos nós. É possível notar que a dificuldade aumenta gradualmente em 10% após a conclusão de cada objetivo. O mapa também inclui duas missões secundárias, representadas pelos nós 303 e 304, que possuem uma baixa taxa de vitória de apenas 10% cada. Para esse cenário é esperado que os agentes encontrem dificuldades nos desafios secundários e acabem desistindo antes de terminar a simulação.

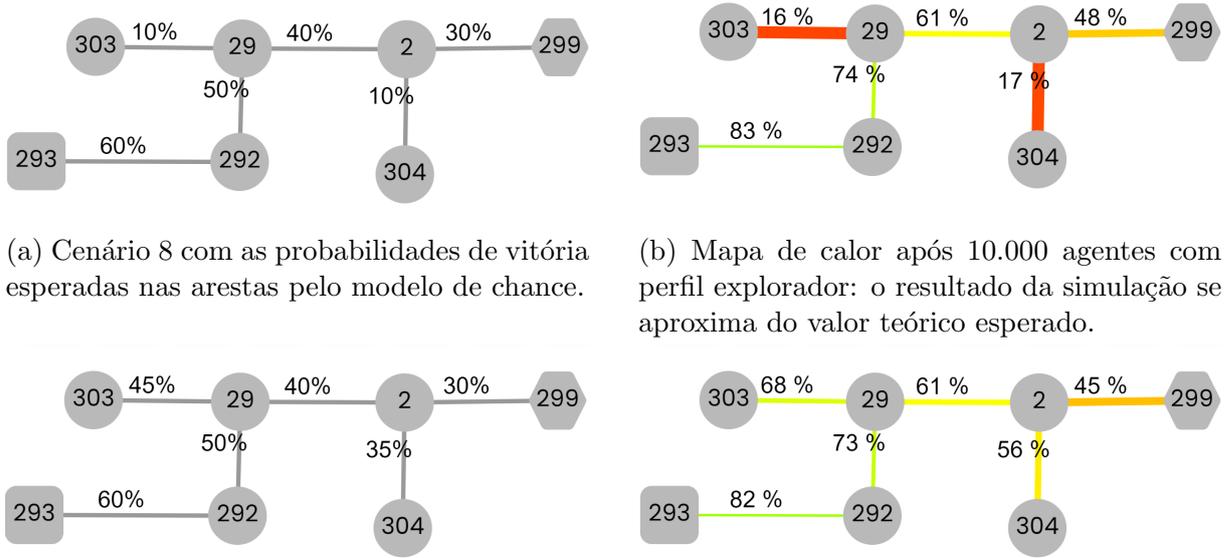


Figura 5.9: Representação gráfica do Cenário 8 com os resultados em mapas de calor. Fonte: do autor.

Na Figura 5.9b, podemos observar os resultados da simulação de 10.000 agentes, onde o percentual de derrotas em relação às tentativas é representado pelas cores e larguras das arestas, enquanto os números nas arestas indicam o percentual de sucesso nos duelos dos agentes.

Ao analisar o grafo, notamos uma grande variação entre o índice de vitória esperado nas arestas e os resultados obtidos. Essa discrepância é resultado do modelo de desistência utilizado, no qual o agente tem duas tentativas para vencer um objetivo. Essa abordagem aumenta a probabilidade de sucesso do agente, permitindo-nos calcular a probabilidade de passar no desafio. Como o agente tem duas tentativas para vencer

um desafio, e essa dupla chance de vitória influencia significativamente a probabilidade de sucesso do agente.

Considerando a aresta entre os nós 303 e 29, com uma probabilidade de vitória de apenas 10%, podemos calcular a chance esperada do agente vencer em duas tentativas. Em cada tentativa, a probabilidade de falha é de 90%. Multiplicando as probabilidades de falha das duas tentativas, obtemos 81%. Portanto, a probabilidade de vitória em pelo menos uma das duas tentativas é de 19%, uma variação de apenas 3% em relação ao valor esperado. Realizando esse mesmo cálculo para o resto das arestas foi observada uma variação de apenas 3% em média.

Ao analisar a Figura 5.9b, podemos observar que os nós 29 e 304 apresentaram as maiores taxas de desistência. Essa análise é importante para identificar os pontos críticos em que os jogadores enfrentaram maior dificuldade, indicando a necessidade de realizar alterações nessas áreas específicas.

Com base nessa análise, foram feitos ajustes manuais nas dificuldades das missões secundárias. Antes dessas modificações, a probabilidade de vitória era de apenas 10% para ambas as missões nos nós 303 e 304. Após os ajustes, como pode ser observado na Figura 5.9c, essas probabilidades aumentaram para 45% e 35%, respectivamente. Agora, há uma diminuição constante na dificuldade ao longo do caminho, proporcionando uma experiência mais equilibrada e aumentando as chances de sucesso dos jogadores nessas etapas do jogo.

Na Figura 5.9d temos o resultado de uma nova simulação conduzida com os mesmos parâmetros estabelecidos por 10.000 iterações, considerando as alterações realizadas. Os resultados dessa simulação demonstram que os agentes simulados encontraram menos dificuldades nos desafios secundários que antes apresentavam um índice de sucesso muito baixo.

5.10 Cenário 9: Ajuste de dificuldade adaptativo

Neste cenário, é demonstrado como o ajuste da dificuldade adaptativo é feito no modelo de desafio por chance fixa e no modelo de desistência por tentativa. O mesmo grafo do cenário de referência é utilizado, mantendo os mesmos parâmetros, mas incorporando a

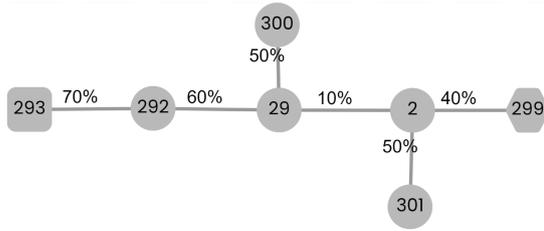
funcionalidade de dificuldade adaptativa. A Figura 5.10a ilustra o grafo específico desse cenário. É possível observar a diferença na aresta entre os nós 29 e 2, onde a dificuldade é significativamente maior em comparação com o restante do grafo. Nesse contexto, espera-se que o modelo de dificuldade em uso ajuste essa dificuldade para se alinhar com as demais arestas do grafo, proporcionando uma experiência de jogo mais consistente.

Na Figura 5.10b temos o resultado de uma simulação de 10.000 agentes. Podemos observar pela cor das arestas o percentual de vitória. Com isso, vemos que a aresta entre os nós 29 e 2 apresenta um índice de vitória significativamente mais baixo em comparação com o restante do grafo.

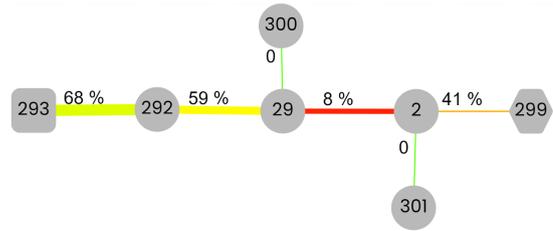
Para abordar essa disparidade, o modelo de dificuldade adaptativa considera o nó com maior taxa de desistência e busca a aresta com o maior número de falhas. A partir dessas informações, é realizada uma média da dificuldade em todo o grafo, juntamente com a média das arestas adjacentes, resultando em um novo ajuste da dificuldade do desafio em questão.

Na Figura 5.10c, é apresentado o novo grafo após o ajuste da dificuldade adaptativa. Podemos observar que a dificuldade do nó entre as arestas 29 e 2 foi alterada para 48. Esse cálculo é determinado pela média da dificuldade do grafo, que é 46, juntamente com a média das arestas ao redor, que é 50. Essa abordagem visa equalizar a dificuldade do desafio com base nas características do grafo na totalidade.

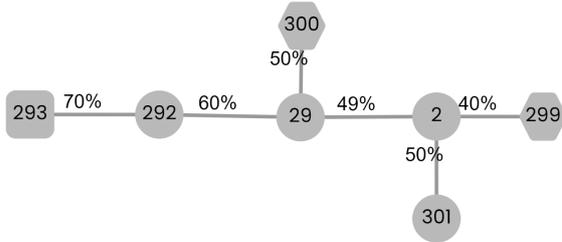
Após o ajuste, foi feita uma segunda simulação com 10.000 agentes com os mesmos parâmetros, mas com a nova dificuldade na aresta ajustada. Na Figura 5.10d temos o resultado desta simulação. Ao analisarmos os índices de vitória representados pelos números e cores das arestas no grafo, observamos um aumento gradual na dificuldade, como esperado. Essa observação valida a eficácia do modelo de dificuldade adaptativa em corrigir uma dificuldade anormal no grafo.



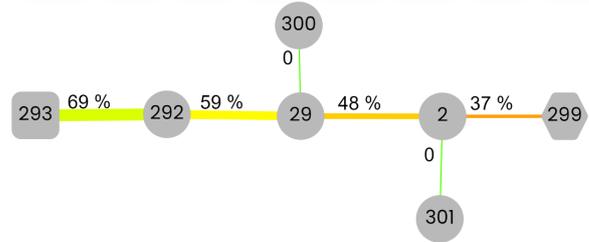
(a) Cenário 9 com a probabilidade de vitória esperada do agente nas arestas pelo modelo de chance.



(b) Mapa de calor após 10.000 agentes com perfil realizador. O resultado da simulação se aproxima do valor teórico esperado.



(c) Cenário 9 após o ajuste do modelo de dificuldade adaptativa.



(d) Mapa de calor após 10.000 agentes da segunda simulação. Observa-se uma diminuição gradual da dificuldade ao longo do mapa pela cor da aresta, após alteração pelo modelo de dificuldade adaptativa.

Figura 5.10: Representação gráfica do Cenário 9 com os resultados em mapas de calor. Fonte: do autor.

5.11 Cenário 10: Ajuste de dificuldade adaptativo (2)

Neste cenário, é explorado o ajuste da dificuldade adaptativa com os modelos de desafio por habilidade e modelo de desistência por *flow*. O mesmo grafo do Cenário 1 é utilizado, porém, com algumas modificações. A habilidade do jogador simulado terá um valor de 1700. A Figura 5.11a ilustra o grafo correspondente a esse cenário.

Após 10.000 agentes com perfil realizador, obtemos o resultado representado na Figura 5.11b. A coloração dos nós indica a quantidade de desistências, enquanto a coloração e os valores das arestas refletem a probabilidade de vitória. Ao analisar a figura, fica evidente que o nó 2 registra o maior número de desistências. Além disso, a aresta que conecta os nós 2 e 29 apresenta uma taxa de vitória de 97%. Considerando o modelo de desistência por tédio adotado, podemos concluir que os jogadores estão desistindo devido ao tédio gerado por essa alta taxa de vitória. Nesse contexto, é esperado que o modelo de dificuldade adaptativa faça um ajuste na mencionada aresta, para diminuir a desistência

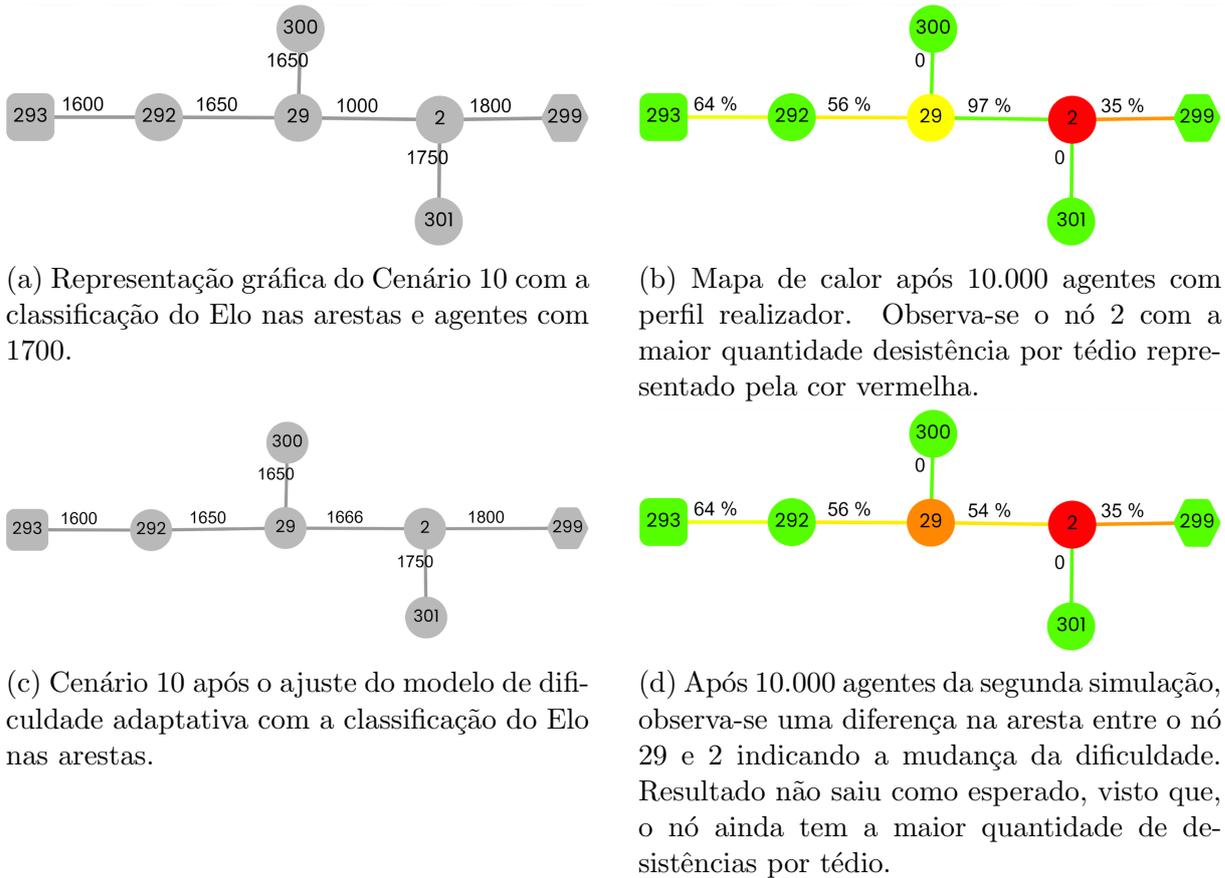


Figura 5.11: Representação gráfica do Cenário 10 com os resultados representados por um mapa de calor. Fonte: do autor.

por tédio no grafo.

Na Figura 5.11c, podemos observar o resultado desse ajuste, que consiste na alteração do valor da aresta entre os nós 2 e 29 de 1000 para 1666. Essa mudança de valor é determinada com base na média das arestas ao redor, juntamente com a estimativa da habilidade do jogador, que foi calculada a partir das vitórias e derrotas, resultando em um valor de 1619.

Finalmente, outra simulação com 10.000 iterações foi realizada. A Figura 5.11d apresenta os resultados obtidos. Nas arestas temos a porcentagem de vitória e os números de desistências nas cores dos nós. Ao analisar a figura, podemos observar a sequência de cores similares nas arestas, o que indica que o ajuste aumentou a dificuldade da aresta comentada. Embora a aresta ajustada agora não apresente uma discrepância na dificuldade tão significativa em relação ao restante do grafo, ainda é evidente que o nó 2 possui a maior quantidade de desistências.

Tende em mente o nó mencionado, fica evidente que o resultado obtido não

está conforme o esperado. Mesmo com o ajuste da dificuldade, o nó 2 continua registrando desistências devido ao tédio gerado pela sequência de desafios fáceis. Essa constatação indica que é necessário revisar e aprimorar o sistema de dificuldade adaptativa para garantir uma experiência mais desafiadora neste caso.

5.12 Considerações Parciais

Após analisar todos os cenários, é possível observar a capacidade do SIMUPRO em representar diferentes tipos de jogos e ambientes de simulação. Cada cenário explorou um aspecto específico, proporcionando diferentes percepções sobre as configurações disponíveis.

Ao utilizar os mapas de calor, foi possível visualizar e analisar os resultados das simulações, auxiliando na compreensão da progressão de dificuldade e na identificação de desafios problemáticos. Essa abordagem visual serve para ajustar o ambiente de simulação para o usuário alcançar os resultados desejados.

No cenário de referência, por exemplo, foi observado o efeito da progressão de dificuldade linear na taxa de sucesso dos jogadores. Isso demonstra a importância de uma progressão gradual e coerente, garantindo que os jogadores enfrentem desafios proporcionais às suas habilidades.

Além disso, os cenários exploraram modelos como o de chance fixa, habilidade e dificuldade adaptativa. Cada um desses modelos oferece abordagens diferentes para ajustar ou representar a dificuldade do jogo com base em probabilidades de vitória, classificação dos jogadores e adaptação dinâmica.

Também foi abordada a questão da desistência por tédio e ansiedade pelo modelo de *flow*, evidenciando a importância de equilibrar a variedade e a dificuldade dos desafios para manter os jogadores engajados. A introdução do modelo de desistência por *flow* mostrou também como a transição suave entre desafios pode afetar a taxa de desistência.

6 Considerações Finais

Este trabalho realizou uma pesquisa exploratória sobre dificuldade adaptativa em jogos digitais, resultando no desenvolvimento de um ambiente de modelagem e simulação representado através da criação de níveis ou fases de jogos. A ferramenta permite que os *level designers* planejem a dificuldade e a navegabilidade dos jogadores, utilizando uma abordagem dinâmica e guiada por meio da simulação de algoritmos de busca em grafos. Além disso, a visualização dos dados feita por mapas de calor destaca as áreas desafiadoras e pontos de desistência dos agentes simulados.

Os objetivos específicos inicialmente propostos foram alcançados em diferentes graus. Foi desenvolvida uma ferramenta de edição de topologia de grafos para representar a estrutura do ambiente de jogo que possibilitou a representação dos jogos. A partir disso, algoritmos de busca foram usados para simular o percurso dos jogadores nos jogos, oferecendo uma análise dinâmica do progresso no jogo. O simulador desenvolvido apresenta elementos configuráveis, permitindo a adaptação do ambiente e suas características. Além disso, a análise dos dados nos grafos por meio de mapas de calor foi desenvolvida, proporcionando uma visualização clara das áreas desafiadoras e dos caminhos percorridos pelos jogadores.

No entanto, é importante mencionar que foram encontradas limitações durante o desenvolvimento do projeto. A representação das arestas foi considerada bidirecional, entretanto, alguns jogos permitem apenas a navegação em um sentido, sendo uma funcionalidade a ser implementada em outra etapa. Adicionalmente, a representação de jogos do tipo *Metroidvania*, com áreas bloqueadas que devem ser liberadas por chaves ou poderes especiais dos personagens, não foi implementada, e o protótipo em sua atual versão não é capaz de representar este tipo de progressão de narrativa. É também uma funcionalidade que permitirá ter modelos mais complexos de simulação, e a representação precisa ser adequada para esses tipos de jogos.

Além disso, o modelo de dificuldade adaptativa implementado não abrangeu completamente todas as áreas do grafo. O modelo apenas ajusta o desafio que apresenta

a maior disparidade do desempenho dos jogadores simulados em relação aos outros desafios do grafo. Outra limitação se deve ao tamanho dos grafos testados e abordados na pesquisa. Como não foram usados grafos de grande porte, isso pode afetar o desempenho da ferramenta bem como os modelos utilizados.

Essas limitações ressaltam a necessidade de estudos adicionais e aprimoramentos futuros para lidar com essas complexidades específicas dos jogos e melhorar o modelo de dificuldade adaptativa, visando uma cobertura mais abrangente e precisa das variações de dificuldade ao longo do jogo. Apesar disso, o presente trabalho representa uma contribuição inicial para o planejamento de jogos digitais, oferecendo uma ferramenta para simulação de níveis, com potencial para melhorar a experiência dos jogadores e diminuir a dependência da experiência prática do *level designer*.

Além disso, é possível expandir a variedade de modelos de jogadores simulados. Atualmente, foram desenvolvidos dois tipos de jogadores (realizador e explorador) para a simulação, mas é possível explorar outras estratégias e comportamentos para representar uma gama mais ampla de estilos de jogo e perfis de jogadores.

Outro aspecto a ser aprimorado é o modelo de adaptação de dificuldade. Embora o modelo implementado neste trabalho tenha considerado as ações do jogador simulado, é importante buscar uma abordagem mais abrangente que permita uma adaptação dinâmica e completa da dificuldade em todas as partes do grafo. Isso contribuirá para uma experiência de jogo mais equilibrada e personalizada para os jogadores.

Além dessas sugestões, outras melhorias podem ser exploradas, como a otimização do desempenho da ferramenta, aprimoramentos na interface do usuário para facilitar a edição e visualização dos grafos, e a implementação de recursos adicionais que auxiliem os designers no planejamento e análise de seus níveis de jogo.

Bibliografia

- ANCEL, M.; HÉRAL, C.; MONTPELLIER, U. *Rayman Legends*. [S.l.]: Ubisoft, 2013.
- APONTE, M.-V.; LEVIEUX, G.; NATKIN, S. Scaling the level of difficulty in single player video games. In: SPRINGER. *International Conference on Entertainment Computing*. [S.l.], 2009. p. 24–35.
- ARAUJO, B. de; FEIJÓ, B. Evaluating dynamic difficulty adaptivity in shoot'em up games. In: *Proceedings of the XII Brazilian Symposium on Games and Digital Entertainment-SBGames*. [S.l.: s.n.], 2013. p. 229–238.
- Atari. *Tempest*. [S.l.]: Atari, 1981. Video Game. Platform: Arcade.
- Atari. *Star Wars*. [S.l.]: Atari, 1983. Video Game. Platform: Arcade.
- Atari. *RoadBlasters*. [S.l.]: Atari, 1987. Video Game. Platform: Arcade.
- BALDWIN, A.; JOHNSON, D.; WYETH, P. A. The effect of multiplayer dynamic difficulty adjustment on the player experience of video games. In: *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. [S.l.: s.n.], 2014. p. 1489–1494.
- BARTLE, R. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD Research*, v. 1, n. 1, p. 1–10, 1996.
- BJÖRK, S.; HOLOPAINEN, J. *Patterns in Game Design*. 1st. ed. Hingham, MA: Charles River Media, 2005.
- CHANEL, G.; REBETEZ, C.; BÉTRANCOURT, M.; PUN, T. Boredom, engagement and anxiety as indicators for adaptation to difficulty in games. In: *Proceedings of the 12th international conference on Entertainment and media in the ubiquitous era*. [S.l.: s.n.], 2008. p. 13–17.
- CHEN, J. Flow in games (and everything else). *Communications of the ACM*, ACM New York, NY, USA, v. 50, n. 4, p. 31–34, 2007.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Introduction to algorithms*. [S.l.]: MIT press, 2009.
- CORTES, C.; VAPNIK, V. Support vector machines. *Machine Learning*, Springer, v. 20, n. 3, p. 273–297, 1995.
- COSTA, P. P. d. Teoria dos grafos e suas aplicações. Universidade Estadual Paulista (Unesp), 2011.
- CSIKSZENTMIHALYI, M. *Flow: The psychology of optimal experience*. [S.l.]: Harper & Row New York, 1990. v. 1990.
- DENISOVA, A.; CAIRNS, P. Player experience and deceptive expectations of difficulty adaptation in digital games. *Entertainment Computing*, Elsevier, v. 29, p. 56–68, 2019.
- ELO, A. E. *The rating of chess players, past and present*. [S.l.]: Arco Pub. Co, 1978.

FEOFILOFF, P.; KOHAYAKAWA, Y.; WAKABAYASHI, Y. Uma introdução sucinta à teoria dos grafos. 2011.

FromSoftware. *Dark Souls*. [S.l.]: FromSoftware, 2011. Video Game. Platform: Various.

HENDRIX, M.; BELLAMY-WOOD, T.; MCKAY, S.; BLOOM, V.; DUNWELL, I. Implementing adaptive game difficulty balancing in serious games. *IEEE Transactions on Games*, IEEE, v. 11, n. 4, p. 320–327, 2018.

HERINGER, V. *Carta Rara de Pokémon É Leiloadada por valor recorde: US\$ 900 mil - nerdbunker*. JovemNerd, 2022. Disponível em: <https://jovemnerd.com.br/nerdbunker/carta-de-pokemon-leiloadada-valor-recorde-us-900-mil/>.

HUNICKE, R.; LEBLANC, M.; ZUBEK, R. Mda: A formal approach to game design and game research. *Proceedings of the AAAI Workshop on Challenges in Game AI*, v. 4, n. 1, 2004.

KENT, S. L. *The Ultimate History of Video Games, Volume 1: From Pong to Pokemon and Beyond... the Story Behind the Craze That Touched Our Lives and Changed the World*. [S.l.]: Crown, 2010. v. 1.

King Digital Entertainment. *Candy Crush Saga*. 2012. Mobile game. Disponível em: <https://www.king.com/game/candycrush>.

Konami. *Super Bomberman 5*. [S.l.]: Konami, 1997. Video Game. Platform: Super Nintendo Entertainment System.

KOSTER, R. *Theory of fun for game design*. [S.l.]: "O'Reilly Media, Inc.", 2013.

LOWOOD, H.; GUINS, R. *Debugging game history: A critical lexicon*. [S.l.]: MIT Press, 2016.

MALONE, T. W. What makes things fun to learn? heuristics for designing instructional computer games. In: *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*. [S.l.: s.n.], 1980. p. 162–169.

MIYAMOTO, S. *The Legend of Zelda*. 1986. Nintendo Entertainment System.

NEEDHAM, M.; HODLER, A. E. *Graph algorithms: practical examples in Apache Spark and Neo4j*. [S.l.]: O'Reilly Media, 2019.

Nintendo. *The Legend of Zelda: A Link to the Past*. [S.l.]: Nintendo, 1991. Video Game. Platform: Super Nintendo Entertainment System.

NINTENDO. *Super Mario Bros. Deluxe*. [S.l.]: Nintendo, 2019.

PACETE, L. G. *Sonic 2 bate recorde de bilheteria, Conheça Outras Adaptações de games*. Forbes, 2022. Disponível em: <https://forbes.com.br/forbes-tech/2022/04/sonic-2-bate-recorde-de-bilheterias-conheca-outras-adaptacoes-dos-games/>.

ROBB, N.; ZHANG, B. Performance-based dynamic difficulty adjustment and player experience in a 2d digital game: A controlled experiment. *Acta Ludologica*, v. 5, n. 1, 2022.

SALEN, K.; ZIMMERMAN, E. *Rules of play: Game design fundamentals*. [S.l.]: MIT press, 2004.

- SCHELL, J. *The Art of Game Design: A book of lenses*. [S.l.]: CRC press, 2008.
- SCHREIBER, I.; ROMERO, B. *Game Balance*. [S.l.]: CRC Press, 2021.
- SEDGEWICK, R.; WAYNE, K. *Algorithms*. [S.l.]: Pearson Education, 2011.
- SILVA, M. P.; SILVA, V. do N.; CHAIMOWICZ, L. Dynamic difficulty adjustment through an adaptive ai. In: IEEE. *2015 14th Brazilian symposium on computer games and digital entertainment (SBGames)*. [S.l.], 2015. p. 173–182.
- STREETON, R.; COOKE, M.; CAMPBELL, J. Researching the researchers: Using a snowballing technique. *Nurse researcher*, Royal College of Nursing Publishing Company (RCN), v. 12, n. 1, p. 35–47, 2004.
- TETRIS. [S.l.]: Various, 1984. Video Game.
- Valve Corporation. *Defense of the Ancients*. [S.l.]: Valve Corporation, 2003. Video Game.
- Valve Corporation. *Left 4 Dead*. [S.l.]: Valve Corporation, 2008. Video Game. Platform: Microsoft Windows, Xbox 360.
- WILLIAMS, D. A brief social history of game play. Lawrence Erlbaum Associates Publishers, 2006.
- WONG, K. W.; FUNG, C. C.; DEPICKERE, A.; RAI, S. Static and dynamic difficulty level design for edutainment game using artificial neural networks. In: SPRINGER. *International Conference on Technologies for E-Learning and Digital Entertainment*. [S.l.], 2006. p. 463–472.
- XUE, S.; WU, M.; KOLEN, J.; AGHDAIE, N.; ZAMAN, K. A. Dynamic difficulty adjustment for maximized engagement in digital games. In: *Proceedings of the 26th International Conference on World Wide Web Companion*. [S.l.: s.n.], 2017. p. 465–471.
- YEE, N. Motivations for play in online games. *CyberPsychology & Behavior*, Mary Ann Liebert, Inc. 2 Madison Avenue Larchmont, NY 10538 USA, v. 9, n. 6, p. 772–775, 2006.
- ZOHAIB, M. Dynamic difficulty adjustment (dda) in computer games: A review. *Advances in Human-Computer Interaction*, Hindawi, v. 2018, 2018.