

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

MARCELO DRUMMOND DA FONSECA

PREVISÃO DE RESULTADO DE COMBATES EM  
DUNGEON AND DRAGONS 5<sup>a</sup> EDIÇÃO  
Uma ferramenta de apoio ao *Game Master*

RIO DE JANEIRO  
2024

MARCELO DRUMMOND DA FONSECA

PREVISÃO DE RESULTADO DE COMBATES EM  
DUNGEON AND DRAGONS 5<sup>a</sup> EDIÇÃO  
Uma ferramenta de apoio ao *Game Master*

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

Orientador: Prof. Geraldo Bonorino Xexéo

RIO DE JANEIRO

2024

## CIP - Catalogação na Publicação

F676p Fonseca, Marcelo Drummond da  
Previsão de resultados de combates em Dungeons  
and Dragons 5a edição / Marcelo Drummond da Fonseca.  
-- Rio de Janeiro, 2024.  
66 f.

Orientador: Geraldo Bonorino Xexéo.  
Trabalho de conclusão de curso (graduação) -  
Universidade Federal do Rio de Janeiro, Instituto  
de Computação, Bacharel em Ciência da Computação,  
2024.

1. Simulação. 2. Dungeons and Dragons. 3. Jogos  
de fantasia. I. Xexéo, Geraldo Bonorino, orient.  
II. Título.


MARCELO DRUMMOND DA FONSECA

PREVISÃO DE RESULTADO DE COMBATES EM  
DUNGEON AND DRAGONS 5<sup>a</sup> EDIÇÃO  
Uma ferramenta de apoio ao *Game Master*

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


Aprovado em 14 de Agosto de 2024

BANCA EXAMINADORA:

Documento assinado digitalmente  
 GERALDO BONORINO XEXEO  
Data: 20/08/2024 08:28:59-0300  
Verifique em <https://validar.iti.gov.br>


---

Geraldo Bonorino Xexéo  
D.Sc. (UFRJ)

Documento assinado digitalmente  
 MARCUS VINICIUS COUTINHO PARREIRAS  
Data: 20/08/2024 08:35:39-0300  
Verifique em <https://validar.iti.gov.br>

---

Marcus Vinícius Coutinho Parreiras  
M.Sc. (CEFET/NI)

Documento assinado digitalmente  
 ADRIANA SANTAROSA VIVACQUA  
Data: 20/08/2024 09:59:52-0300  
Verifique em <https://validar.iti.gov.br>

---

Adriana Santarosa Vivacqua  
D.Sc. (UFRJ)

A meus pais, que me deram todas as oportunidades e apoio que precisei para chegar tão longe.

## AGRADECIMENTOS

Agradeço a meu orientador Geraldo Xexéo, por me guiar ao longo desse árduo processo e mostrar o caminho que o projeto precisava tomar. Muito obrigado.

A Marcus Parreiras, cujas sugestões e apoio constantes foram necessárias para dar forma ao projeto. Muito obrigado.

A minha amiga Sandra Vasconcelos, por seu auxílio em questões de *design* e por acreditar no projeto. Muito obrigado.

A meus pais, por sempre me darem apoio e motivação para progredir com o trabalho. Muito obrigado.

## RESUMO

O trabalho apresenta a criação de uma ferramenta para simulação de encontros de combate na quinta edição de *Dungeons & Dragons*. Por muitos anos, *Game Masters* criaram encontros de combate utilizando as regras ditadas nos livros oficiais, sem possuir uma compreensão maior dos impactos dos desafios que desejam apresentar para seus jogadores. O uso de simulação feita baseada nas regras oficiais de *Dungeons & Dragons* permite a observação de centenas ou milhares de possíveis resultados para esses desafios, possibilitando ao *Game Master* uma análise do cenário que construiu antes de apresentá-lo aos outros jogadores, tendo assim maior controle sobre os eventos do jogo. A ferramenta foi verificada com análise quantitativa, comparando os resultados obtidos pelas simulações com resultados esperados pelas regras de construção de encontros encontradas nos livros oficiais; e com análise qualitativa, com entrevistas e questionários respondidos por jogadores experientes. Os resultados foram positivos, mostrando que a ferramenta possui acurácia em sua simulação e utilidade para a preparação de encontros de combate, assim conseguindo cumprir seu propósito.

**Palavras-chave:** simulação; Dungeons and Dragons; jogos de fantasia.

## ABSTRACT

The work presents the creation of a tool for simulating combat encounters using the ruleset of the fifth edition of Dungeons & Dragons. For many years, game masters have created combat encounters using only the rules dictated in the official books, without a deeper understanding of the challenges they present to their players. The use of simulation created using the official ruleset of Dungeons and Dragons allows for the observation of hundreds or thousands of possible outcomes, permitting the game master to analyze and have better control of the scenario their players will face during the game. The tool was verified by using quantitative and qualitative analysis, with the quantitative analysis taking the form of a comparison of the results obtained by simulation with the expected difficulty of an encountered as defined by the book, and the qualitative analysis taking the form of interviews and questionnaires answered by experienced players. The results were positive, showing that the tool is accurate in the results of its simulation and that it is useful for the preparation of combat encounters, thus fulfilling its purpose.

**Keywords:** simulation; Dungeons and Dragons; role playing game.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Livros principais do Dungeons and Dragons 5ª Edição . . . . .	13
Figura 2 – O diagrama de classes do aplicativo. . . . .	35
Figura 3 – Tela exemplificando uso do simulador para uma única luta. . . . .	37
Figura 4 – Tela exemplificando uso do simulador para lutas em sequencia. . . . .	38
Figura 5 – Arquivo <i>Results</i> com as métricas simples obtidas pela simulação. . . . .	38
Figura 6 – Arquivo <i>Battle Log</i> descrevendo cada batalha passo a passo incluindo suas métricas. . . . .	39
Figura 7 – Métricas totais devolvidas no final do arquivo <i>Battle Log</i> . . . . .	40
Figura 8 – Tela de edição dos atributos dos atores. . . . .	40
Figura 9 – Enter Caption . . . . .	41
Figura 10 – Tela de edição das ações dos atores. . . . .	42
Figura 11 – Tela para adicionar condições permanentes aos atores. . . . .	43
Figura 12 – Tela para edição de recursos disponíveis aos atores. . . . .	44
Figura 13 – Tela para edição de combos dos atores. . . . .	45
Figura 14 – Tela de edição dos atributos base da ação. . . . .	46
Figura 15 – Tela de edição dos custos das ações. . . . .	47
Figura 16 – Tela para edição dos efeitos das ações. . . . .	48
Figura 17 – Tela para edição dos atributos básicos de condições . . . . .	49
Figura 18 – Tela para edição dos modificadores de ataque da condição . . . . .	50
Figura 19 – Tela para edição dos modificadores de defesa da condição . . . . .	51
Figura 20 – Tela para edição dos modificadores da economia de ações da condição. . . . .	52
Figura 21 – Tela para edição de efeitos da condição que ocorrem ao longo do tempo. . . . .	53
Figura 22 – Tela para definição de nomes de recursos. . . . .	54
Figura 23 – Duração de simulação sem logs detalhadas . . . . .	55
Figura 24 – Duração de simulação com logs detalhadas . . . . .	56
Figura 25 – Partida espelho entre diversos tipos de criaturas diferentes. . . . .	57

## LISTA DE TABELAS

Tabela 1 – Resultados da avaliação com encontros criados manualmente . . . . .	59
Tabela 2 – Resultados da avaliação com encontros gerados aleatoriamente . . . . .	61

## LISTA DE ABREVIATURAS E SIGLAS

RPG	Role-Playing Game
D&D	Dungeons & Dragons
GM	Game Master
XdY+Z	X dados de Y lados com modificador Z

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>12</b>
<b>2</b>	<b><i>DUNGEONS AND DRAGONS</i> . . . . .</b>	<b>14</b>
2.1	O QUE É UM RPG . . . . .	14
2.2	O JOGO <i>DUNGEONS AND DRAGONS</i> . . . . .	14
2.3	O COMBATE EM D&D . . . . .	15
<b>3</b>	<b>TRABALHOS CORRELATOS . . . . .</b>	<b>19</b>
<b>4</b>	<b>PROPOSTA . . . . .</b>	<b>21</b>
4.1	CALCULANDO MATEMATICAMENTE . . . . .	21
4.2	SIMULAÇÃO . . . . .	22
4.3	MÉTRICAS DE BALANCEAMENTO . . . . .	27
4.4	ESCOPO DO PROJETO . . . . .	28
4.5	REQUISITOS FUNCIONAIS . . . . .	28
4.6	REQUISITOS NÃO-FUNCIONAIS . . . . .	29
4.7	AVALIAÇÕES . . . . .	30
4.7.1	<b>Avaliações Quantitativas . . . . .</b>	<b>30</b>
4.7.2	<b>Avaliações Qualitativas . . . . .</b>	<b>30</b>
4.7.2.1	Escalas de Likert . . . . .	30
<b>5</b>	<b>IMPLEMENTAÇÃO . . . . .</b>	<b>31</b>
5.1	FERRAMENTAS UTILIZADAS . . . . .	31
5.2	EVOLUÇÃO DO SIMULADOR . . . . .	31
5.2.1	<b>Versão 1 . . . . .</b>	<b>31</b>
5.2.2	<b>Versão 2 . . . . .</b>	<b>32</b>
5.2.3	<b>Versão 3 . . . . .</b>	<b>32</b>
5.2.4	<b>Versão 4 . . . . .</b>	<b>33</b>
5.2.5	<b>Versão 5 . . . . .</b>	<b>34</b>
5.2.6	<b>Versão 6 . . . . .</b>	<b>34</b>
5.2.7	<b>Versão 7 . . . . .</b>	<b>35</b>
5.3	ARQUITETURA FINAL . . . . .	35
5.3.1	<b>Diagrama de Classes . . . . .</b>	<b>35</b>
5.4	INTERFACE/ COMO FUNCIONA . . . . .	36
5.4.1	<b>Simulator . . . . .</b>	<b>36</b>
5.4.1.1	Results e Battle Log . . . . .	38

<b>5.4.2</b>	<b>Creature</b> . . . . .	<b>40</b>
5.4.2.1	Resistances . . . . .	41
5.4.2.2	Actions . . . . .	42
5.4.2.3	Conditions . . . . .	43
5.4.2.4	Resources . . . . .	44
5.4.2.5	Combos . . . . .	44
<b>5.4.3</b>	<b>Action</b> . . . . .	<b>45</b>
5.4.3.1	Action Stats . . . . .	45
5.4.3.2	Costs . . . . .	46
5.4.3.3	Effects . . . . .	47
<b>5.4.4</b>	<b>Condition</b> . . . . .	<b>47</b>
5.4.4.1	Condition Stats . . . . .	47
5.4.4.2	Offense Modifiers . . . . .	48
5.4.4.3	Defense Modifiers . . . . .	49
5.4.4.4	Economy Modifiers . . . . .	50
5.4.4.5	Over Time . . . . .	50
<b>5.4.5</b>	<b>Resources</b> . . . . .	<b>50</b>
<b>6</b>	<b>AVALIAÇÃO</b> . . . . .	<b>55</b>
6.1	AVALIAÇÃO DA API . . . . .	55
<b>6.1.1</b>	<b>Testes de Desempenho</b> . . . . .	<b>55</b>
<b>6.1.2</b>	<b>Testes de Sanidade</b> . . . . .	<b>56</b>
6.1.2.1	Partida Espelho . . . . .	56
6.1.2.2	Combate altamente desbalanceado . . . . .	56
6.1.2.3	Combate com ator imune . . . . .	57
<b>6.1.3</b>	<b>Comparação com o Modelo Oficial do Dungeons &amp; Dragons</b> .	<b>57</b>
6.1.3.1	Utilizando encontros definidos manualmente . . . . .	57
6.1.3.2	Utilizando encontros definidos de forma aleatória . . . . .	60
6.2	AVALIAÇÃO QUALITATIVA COM JOGADORES . . . . .	61
<b>7</b>	<b>CONCLUSÃO</b> . . . . .	<b>63</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>64</b>

## 1 INTRODUÇÃO

*Dungeons and Dragons* (D&D) é um *Tabletop Role-Playing Game* (TTRPG) originalmente publicado em 1974 que permite que um grupo de jogadores criem personagens para enfrentar desafios apresentados por um jogador tomando papel de mestre do mundo, denominado Dungeon Master (DM). De acordo com a *Wizards of the Coast*, mais de 50 milhões de pessoas mundialmente já jogaram D&D, com a edição atual, a quinta edição, possuindo mais de 10 milhões de jogadores ativos.

Uma dificuldade comum encontrada pelos DMs quando planejando as aventuras para seus jogadores está em criar encontros de combate equilibrados e interessantes para seus jogadores, devido à complexidade dos personagens e seus inimigos e a quantidade de interações e conhecimento do jogo necessária. Assim, esse trabalho busca apresentar o desenvolvimento de uma ferramenta capaz de auxiliar DMs com esse desafio mediante simulação, considerando as regras da quinta edição de D&D apresentadas nos livros oficiais *Player's Handbook*, *Dungeon Master's Guide* e *Monster Manual* (Wizards RPG Team, 2014c; Wizards RPG Team, 2014a; Wizards RPG Team, 2014b).

O objetivo do trabalho é criar uma ferramenta capaz de simular diversos encontros de combate da quinta edição de D&D, com personagens e inimigos customizáveis, seguindo as regras oficiais descritas nos livros principais. Sub-objetivos incluem a criação de uma interface para facilitar a customização dos diversos aspectos dos personagens e organizar os encontros de combate; e a devolução de métricas úteis para os propósitos do DM.

É esperado que os resultados obtidos por simulação estejam alinhados com os resultados dos sistemas descritos nos livros oficiais, condição que foi verificada por testes comparando ambos. Além disso, foi feita uma avaliação qualitativa da ferramenta por jogadores de D&D experientes, mostrando que a ferramenta é útil.

A partir desta introdução, o trabalho será desenvolvido em mais seis seções. A seção 2 descreve a fundo D&D e o como funciona seu sistema de combate; a seção 3 descreve os métodos e ferramentas utilizadas durante o trabalho; a seção 4 define a proposta final do projeto; a seção 5 detalha a implementação do simulador; a seção 6 mostra as avaliações realizadas e seus resultados; e a seção 7 apresenta as conclusões finais do trabalho.

Figura 1 – Livros principais do Dungeons and Dragons 5ª Edição



Fonte: <http://dnd5ed.github.io/>

## 2 DUNGEONS AND DRAGONS

Neste capítulo, serão descritos conceitos importantes para compreender o trabalho, incluindo o que é um RPG (*Role Playing Game*), e mais especificamente o como funciona o jogo que é o foco deste trabalho, *Dungeons and Dragons*.

### 2.1 O QUE É UM RPG

RPG (*Role Playing Game*) é uma categoria de jogos digitais ou de mesa que se popularizou a partir da década de 1970. O termo indica um jogo focado em história e personagens, seu desenvolvimento ao longo do tempo e os desafios que tem que superar. Frequentemente, RPGs incluem combate em estilo de turnos, com os jogadores alternando o turno com os inimigos e cenário de fantasia medieval. Essa mecânica de combate se popularizou devido a alguns dos jogos que foram fundamentais em tornar o gênero famoso, como *Dragon Quest*, *Final Fantasy* e o próprio *Dungeons and Dragons* original (HITCHENS; DRACHEN, 2008).

### 2.2 O JOGO DUNGEONS AND DRAGONS

*Dungeons and Dragons* é uma série de jogos de mesa de RPG, com sua primeira edição publicada em 1974, e a edição mais recente, a quinta edição, publicada em 2014. Em 2024 e 2025, serão lançados os livros para a próxima edição, nomeada *One D&D*. As regras essenciais do jogo são disponibilizadas em 3 livros: O *Player's Handbook* (Wizards RPG Team, 2014c), que é utilizado para a criação de personagens; o *Dungeon Master's Guide* (Wizards RPG Team, 2014a), que inclui as regras gerais e o como o DM deve conduzir o jogo; e o *Monster Manual* (Wizards RPG Team, 2014b), que é uma coleção dos tipos de criaturas e monstros que os jogadores podem encontrar, incluindo as regras para cada uma delas.

O jogo é criado para ser jogado em um grupo, tipicamente de 5 pessoas, em que uma toma a função de DM (*Dungeon Master*), que controla o mundo, decidindo quais eventos acontecem, controla os NPCs (*Non-Player Characters*) e indica aos personagens os resultados de suas ações.

Os outros jogadores criam um único personagem, usando os sistemas definidos no *Player's Handbook*, e utilizam esses personagens para interagir com o mundo, em algum tipo de aventura. Isso quase sempre envolve combate contra outros grupos e/ou monstros, que são detalhados no *Monster Manual*.

Além do conteúdo criado diretamente pela *Wizards of the Coast*, existe um volume enorme de conteúdo criado por terceiros, ambos publicados pela *Wizards of the Coast*,



como foi o caso de *Explorer's Guide to Wildemount*(MERCER, 2020), ou publicados separadamente, como os produtos da *Kobold Press*, que publicou livros como *Creature Codex for 5th Edition*(PRESS, 2018) para serem usados no jogo. Além disso, existem inúmeras publicações criadas por fans do jogo e disponibilizadas online, como no site *Dungeon Master's Guild*<sup>1</sup>.

## 2.3 O COMBATE EM D&D

Os personagens e monstros tem algumas características em comum

- Nível (*Level*), medida numérica indicando a força, chamado *Challenge Rating* para monstros; varia de 1 a 20 para os jogadores e de 0 a 30 para monstros;
- Pontos de Vida (*Hit Points*), A quantidade de dano que precisam tomar para ser derrotados;
- Classe de Armadura (*Armor Class*), O quão difícil são de acertar;
- Iniciativa (*Initiative*), O quão rápido consegue agir em comparação a outros;
- Bônus de Proficiência (*Proficiency Bonus*), Valor que pode variar de 2 até 9 dependendo do Level, com múltiplas funções;
- Deslocamento (*Speed*), Valor indicando a quantidade de espaços que um personagem consegue se mover em um turno), e
- Atributos (*Ability Score*), Os atributos básicos que determinam os valores de múltiplas outras propriedades. Varia de 1 até 30, mas personagens de jogadores geralmente ficam entre 8 e 20).
  - Força (*Strength*), Indica a força física do personagem)
  - Destreza (*Dexterity*), Indica a agilidade do personagem)
  - Constituição (*Constitution*), Indica a resistência a dano do personagem)
  - Inteligência (*Intelligence*), Indica a capacidade do personagem de memorizar informação e realizar deduções lógicas)
  - Sabedoria (*Wisdom*), indica a capacidade do personagem de observar e compreender o mundo a sua volta, e
  - Carisma (*Charisma*), a capacidade do personagem de influenciar outros.

<sup>1</sup> *Dungeon Master's Guild* é um site disponível em <https://www.dmsguild.com/> em que DMs podem publicar conteúdo personalizado para ser utilizado por outros DMs. Ultimo acesso em 2024-07-23

Além dessas propriedades comuns, há muitas outras estabelecidas na criação do personagem e nos monstros já definidos, incluindo ataques, magias ou equipamentos.

Quase toda ação em D&D requer que os jogadores ou DM rolem dados para determinar se a ação tem sucesso ou não, e para algumas outras funcionalidades como definir o dano dado por um ataque. É utilizada a notação  $XdY+Z$ , que significa “role  $X$  dados de  $Y$  lados e adicione  $Z$  ao resultado”, então, por exemplo,  $2d6+5$  significa “Role 2 dados de 6 lados e adicione 5 ao resultado”. O dado utilizado mais frequentemente é o d20, que é utilizado para ataques, testes de habilidade e testes de resistência. Para o d20 especificamente, existe o conceito de vantagem e desvantagem, que indicam que o personagem tentando a rolagem tem uma chance maior ou menor de ter sucesso devido às condições atuais da situação. Para rolar com vantagem, ou desvantagem, o d20 é rolado duas vezes, e somente o maior, ou menor, resultado é considerado, respectivamente.

O combate em D&D ocorre em turnos, com cada jogador tendo sua própria vez de jogar. No início do combate, todos os jogadores rolam 1d20 e adicionam seu bônus de iniciativa para determinar a ordem em que os personagens vão agir, e o DM faz o mesmo para todos os inimigos. Uma rodada do jogo consiste em cada personagem e inimigo tomando um turno, na ordem de maior iniciativa até menor iniciativa, e repetindo isso até que todos os jogadores ou inimigos sejam derrotados. Como ambos os personagens de jogadores e os monstros seguem, em sua maioria, as mesmas regras fundamentais para combate, ambos serão denominados como atores. No turno de um ator, ele tem direito a movimentação, uma ação, e uma ação bônus, detalhadas a seguir. Ele também tem uma reação, que pode ser usada fora do seu turno e tem regras específicas.

O mapa é mostrado como uma grade de quadrados, em que cada quadrado tem lado de 1,5m. Para se mover para um quadrado adjacente, o ator gasta 1,5m de seu movimento, e pode fazer isso até chegar no limite do seu valor de deslocamento. Além do deslocamento comum, existem outros tipos de deslocamento, como voo, nado, escalada e escavada, que seguem as mesmas regras, mas permitem deslocamento por diferentes tipos de terreno. É possível existirem obstáculos que impedem o movimento dos atores, incluindo terreno difícil, que dobra o gasto de movimento.

A ação é a mais complexa das opções que um ator pode realizar em seu turno, e também geralmente a mais poderosa. Durante seu turno, um ator pode tomar uma das seguintes ações:

- Atacar (*Attack*), em que o ator escolhe um de seus ataques e rola  $1d20 +$  seu bônus de ataque, que é calculado como o seu bônus de atributo (geralmente de força ou destreza) + seu bônus de proficiência + qualquer outro bônus adicionado por magias, armas especiais ou outros efeitos específicos. Se esse valor for maior ou igual à classe de armadura do alvo, o ataque acerta, e o ator rola dano, que é dado por uma quantidade de dados determinada pelo ataque + seu bônus de atributo. Esse

valor é então reduzido dos pontos de vida do alvo.

- Magia (*Cast a Spell*), em que o ator utiliza uma de suas magias, cujas regras são específicas para cada. Geralmente, magias ofensivas ou que causam alguma condição em um oponente necessitam de uma rolagem de ataque, como descrito na ação Atacar, mas utilizando o atributo de magia do personagem (inteligência, sabedoria ou carisma, dependendo da classe), ou com frequência maior, força o oponente a realizar uma salvaguarda de algum atributo específico (o atributo específico depende da magia). Para realizar uma salvaguarda, o alvo da magia rola 1d20 + sua salvaguarda. Se o valor obtido for maior ou igual à CD (Classe de Dificuldade) de magia do ator utilizando a magia, o alvo “passa” na salvaguarda, e o efeito da magia é reduzido ou nulificado. Caso contrário, o alvo recebe o efeito completo. O CD de magia de um ator é calculado como  $8 + \text{seu bônus de atributo de magia} + \text{seu bônus de proficiência}$ .
- Correr (*Dash*), em que o ator pode se movimentar novamente, até o máximo de seu deslocamento.
- Desviar (*Dodge*), em que o ator causa desvantagem em todos os ataques contra ele até o fim do turno.
- Desengajar (*Disengage*), em que o ator evita ataques de oportunidade até o fim do turno.
- Preparar uma ação (*Ready an Action*), em que o ator declara sua intenção de realizar uma ação depois de seu turno, e prepara para realizar essa ação como uma reação.
- Ajudar (*Help*), em que o ator dá vantagem para um de seus aliados no próximo ataque ou teste de habilidade que fizer.
- Fazer um teste de habilidade (*Make an Ability Check*), em que o ator faz um teste de habilidade com alguma de suas habilidades. As mais comuns em combate incluem furtividade (*Stealth*), para tentar se esconder; percepção (*Perception*), para encontrar um indivíduo escondido; e atletismo (*Athletics*), para tentar empurrar, agarrar ou fazer outro ator cair no chão.
- Outros, que incluem ações de regras alternativas, regras não oficiais, e as múltiplas ações únicas e específicas de cada ator, determinada pelo como o ator foi feito.

A ação bônus é alguma outra ação, geralmente menor, que um ator pode realizar em seu turno além de sua ação normal. Não existem ações bônus genéricas, que qualquer ator pode usar, elas são específicas para cada personagem e monstro.

Finalmente, tem a reação, que é uma ação especial que pode ser feita fora do turno do personagem, quando alguma condição é cumprida. As duas reações utilizáveis por

qualquer ator são a ação determinada pela ação “Preparar uma ação” e o ataque de oportunidade, que é um ataque que pode ser feito com as mesmas regras da ação “Atacar” quando um oponente tenta sair do alcance do ator sem desengajar antes. Existem outras reações que os atores podem ter acesso, dependendo de suas habilidades individuais.

### 3 TRABALHOS CORRELATOS

Foi realizada uma revisão rápida da literatura, utilizando para a consulta os descritores RPG, *Dungeons and Dragons*, D&D, *COMBAT* e *ENCOUNTER* na combinação (RPG OR “Dungeons and Dragons” OR “D&D”) AND (COMBAT OR ENCOUNTER). A busca foi feita nas bases SCOPUS, SOL e Google Scholar, sem restrição de idioma ou data. Não foram obtidos artigos relevantes nas primeiras duas bases, mas existiam alguns no Google Scholar, dos quais os mais relevantes são descritos a seguir.

No trabalho “*Automatic Play-Testing of Dungeons and Dragons Combat Encounters*” (SHYNE, 2023), os autores buscaram criar um ambiente virtual em que DMs possam simular encontros de combate usando agentes virtuais para prever dificuldade. Em seguida, foi feita uma pesquisa com DMs reais, utilizando questionários, para saber se os resultados das simulações tiveram acurácia maior, quando comparado às regras oficiais de D&D.

Na tese “*D&D 5E Mass Combat Simulation*” (OHNEMUS, 2019), o autor buscou realizar simulações de combate utilizando uma escala significativamente maior que o normal, sem alterar as mecânicas do jogo.

Em “*FIREBALL: A Dataset of Dungeons and Dragons Actual-Play with Structured Game State Information*” (ZHU et al., 2023), os autores apresentam um banco de dados de diálogos e estados de jogo tomado de sessões reais de D&D, mostrando que quando usado por um *Large Language Model* (LLM), como GPT-3, pode melhorar a geração de linguagem natural, de acordo com métricas humanas e automatizadas. Outro trabalho próximo a esse assunto foi “*Dungeons and Dragons as a Dialog Challenge for Artificial Intelligence*”, em que um LLM foi treinado a partir de bancos de dados semelhantes ao FIREBALL para conseguir gerar turnos, produzir diálogo e tomar conta do estado atual do jogo, obtendo resultados de qualidade melhor que sistemas de linguagem não treinados, mas ainda tendo dificuldade em tomar conta do estado do jogo.

No trabalho “*TaRCoS: Providing a Combat Simulation Tool for Tabletop Role-Playing Games*” (JUNIOR; SARINHO, 2021), os autores criaram uma ferramenta capaz de simular encontros de combate em jogos de RPG, com o objetivo de minimizar o tempo gasto em combate e prover dados sobre o combate. A ferramenta foi feita levando em conta múltiplos possíveis sistemas de RPG diferentes, cada um com suas próprias regras, sendo necessárias certas abstrações nas simulações.

Os autores de “*Life’s a Party: Analyzing Party Compositions for Combat Encounters in Dungeons & Dragons Fifth Edition through Monte Carlo Methods*” (HAUGE; STARR, 2021)” criaram uma simulação simples de combate em D&D, com o objetivo de estudar quais combinações de classes de personagens criam um grupo mais efetivo. Para tanto, realizaram simulações de combate utilizando cada combinação de classes possível para

grupos de quatro jogadores, contra grupos pré-definidos de monstros, e armazenaram os resultados de cada combate para comparação.

Foram exploradas também ferramentas disponíveis online e literatura cinzenta, com as mais relevantes sendo descritas a seguir.

A ferramenta “*D&D 5e Encounter Simulator*” (FERLA, ) utiliza um simulador escrito em Python para realizar 1000 simulações de combate, com os combatentes definidos pelo usuário ou escolhidos de opções pré-definidas.

“*The Finished Book*” (EDUNN, ) contém múltiplas análises de diferentes aspectos de D&D, inclusive sobre cálculos de dificuldade, reduzindo as diferentes propriedades de jogadores e monstros a valores simples de ofensiva e defensiva, e mostrando como é possível utilizar esses valores para comparar a diferença de força entre os dois lados do combate.

De maneira geral, a literatura revisada utilizou principalmente simulação para tentar obter informações sobre o nível de dificuldade de combates em D&D, conferindo os resultados obtidos com outras fontes como encontros reais. Com essa informação, os DMs conseguem fazer decisões melhores em sua construção de encontros, criando um jogo mais balanceado e levando a uma experiência melhor para os jogadores.

Outros trabalhos de menor relevância sobre o jogo foram revisados, mas foram principalmente relacionados a análise de sentimentos e efeitos psicológicos de D&D e não aos encontros de combate em si.

## 4 PROPOSTA

Será definida a seguir a proposta do como encontrar um valor para a dificuldade dos encontros de combate. Será analisado inicialmente o caso mais simples, e mais complexidade será adicionada passo a passo.

Existem dois métodos que podem ser utilizados para encontrar esse valor: calculá-lo matematicamente, ou realizar uma simulação com as regras de D&D. Ambos métodos tem seus méritos, e é de interesse observar uma comparação entre os dois métodos.

O caso mais simples possível, para ambos métodos, seria uma luta entre dois atores com apenas ataques simples e sem nenhuma habilidade especial lutando um contra o outro. Nesse caso, é necessário saber, para cada um, apenas a quantidade de pontos de vida, classe de armadura, probabilidade de acertar e o dano de seu ataque. A seguir são adicionados outros níveis de complexidade como ataques em área, múltiplos combatentes, habilidades especiais, entre outras.

### 4.1 CALCULANDO MATEMATICAMENTE

Para analisar matematicamente, podemos calcular um valor “dificuldade”, dado como a razão entre a quantidade de turnos necessária para os inimigos serem derrotados e a quantidade de turnos necessária para os jogadores serem derrotados. Para isso, poderíamos reduzir os atores a apenas suas características ofensivas e defensivas, e compará-las entre si, sendo a dificuldade tratar as características para conseguir reduzi-las a valores simples de ataque e defesa.

Em “*The Finished Book*”, são feitas diversas análises matemáticas de elementos de D&D, incluindo cálculos de “HP efetivo” considerando os pontos de vida e classe de armadura; “dano por turno efetivo” considerando o dano e chance de ataques acertarem; e valores de “XP efetiva” calculados a partir desses dois valores, sendo este um valor indicativo da força real de um monstro e portanto o desafio que trás.

Uma possibilidade seria calcular individualmente o dano por turno médio que um ator possui para cada inimigo, utilizando os valores reais de dano, bônus de ataque e classe de armadura para o grupo. Em seguida, é feita uma média simples desses valores, ou uma média ponderada com porcentagens específicas para cada inimigo. Assim seria obtido um valor para o dano por rodada médio de cada ator, e a quantidade de pontos de vida total dos inimigos dividida pela soma desse valor para cada ator no time aliado seria a quantidade de rounds necessários para o time aliado derrotar o time inimigo, e vice-versa. Essa análise ainda precisa ser mais detalhada, para incluir alguns aspectos específicos como vantagem/desvantagem, ataques que afetam múltiplos inimigos, e condições negativas ou positivas. Alguns desses fatores são mais fáceis de visualizar em simulação. Outra

desvantagem desse método é que não é tão efetivo quando tratando de combates em que um lado precisa de pouco dano para derrotar o outro, devido ao como um ataque envolve ambas uma chance binária de acerto ou erro e uma rolagem de dados para definir o dano, com o objetivo principal de fazer os pontos de vida do inimigo chegarem a zero podendo ter probabilidade diferente em relação às médias.

## 4.2 SIMULAÇÃO

É possível simular combates reais usando as propriedades dos atores conforme as regras oficiais de D&D. Em seguida, seriam armazenadas as informações obtidas após a luta ser simulada múltiplas vezes e analisados os resultados. Nesse caso, a dificuldade se trataria de implementar as regras do jogo e a lógica das ações de cada ator.

Para simular, são primeiramente carregadas as informações de cada ator e seus times. É rolada iniciativa para cada ator, e o combate progride na ordem de turnos da iniciativa mais alta até a mais baixa, com regras simples para a lógica de ações tomadas por cada personagem. Em termos de alvos, pode-se deixar o usuário da ferramenta definir as regras que cada ator segue, ou podemos implementar uma lógica simples, dizendo, por exemplo, “Atacantes de curta distância lutam contra outros atacantes de curta distância primeiro, e depois vão atrás dos de longa distância”, “Atacantes de longa distância podem atacar qualquer inimigo”, “Personagens com habilidades de cura focam em manter seus aliados vivos”, entre outras regras. Deixar o usuário decidir qual lógica os atores seguem é de interesse, para poder observar como as chances de vitória e derrota são afetadas pelo planejamento dos jogadores.

Após definido o alvo e as ações a serem tomadas pelos atores, é necessário que se rolem dados para saber seus resultados. É possível que errem completamente; acertem um ataque; curem um aliado, tudo definido pelos valores obtidos pelos dados. Essas pequenas variações podem levar a resultados muito diferentes ao longo dos turnos, portando é importante realizar a simulação múltiplas vezes para obter confiança maior e apresentar resultados com maior acurácia ao usuário.

Assim, a simulação pode ser realizada seguindo o algoritmo descrito no algoritmo 1. O turno de um ator pode ser dividido em três etapas: o início do turno, as ações e o fim do turno.

- Início do turno.
  - Recuperar recursos de recarga por turno, caso existam.
  - Atualizar duração de condições relevantes.
- Ações.
  - Decidir quais ações utilizar em quais alvos.



```

Input:  $T_1$  e  $T_2$ , duas listas de atores
Output: 1 ou 2, o id do time vencedor
for  $Ator$  in  $T_1, T_2$  do
  |  $listaIniciativas \leftarrow calcularIniciativa(Ator)$ 
end
 $atoresOrdenados \leftarrow gerarAtoresOrdenados(listaIniciativas, T_1, T_2);$ 
while  $T_1$  e  $T_2$  possuem atores vivos do
  | for  $i = 0$  to  $tamanho(atoresOrdenados)$  do
  | | if  $atoresOrdenados[i]$  está vivo then
  | | |  $atoresOrdenados[i]$  realiza seu turno
  | | end
  | end
end
 $id \leftarrow 1$  se todos os membros do time 2 estiverem mortos e 2 caso contrário;
return  $id$ 

```

**Algorithm 1:** Algoritmo da simulação

- Realizar as ações.
- Fim do turno.
- Atualizar duração de condições relevantes.

É importante atualizar condições ambas no começo e no fim do turno do ator, pois existem condições que duram até o começo do turno e condições que duram até o fim do turno. Há também condições cuja duração é atualizada no turno do ator que aplicou a condição e condições cuja duração é atualizada no turno do ator que foi o alvo da condição, então há diversos cenários diferentes a serem considerados.

A parte crucial para o funcionamento da simulação são as ações. O ator deve decidir quais ações quer tomar contra quais alvos, e em seguida executá-las. Isso levará aos algoritmos de decisão e das ações. Será assumido que o ator sempre buscará utilizar as ações que dão o maior benefício para seu time e/ou prejudicam o time oposto da maior forma. Para calcular o valor de uma ação, pode ser utilizado um sistema de pontos, em que os pontos são definidos diferentemente para cada tipo de ação.

Para ações que causam dano, que é a maioria das ações utilizadas em D&D, é importante saber dois aspectos: A chance do efeito funcionar, e o dano médio causado. A chance de acertar um ataque pode ser dada pela equação:

$$HIT = \frac{\min(19, \max(1, 21 - (CA - BA)))}{20} \quad (4.1)$$

Em que HIT é a chance de acertar, CA é a classe de armadura do alvo e BA é o bônus de ataque do atacante. Note que o valor não pode ser menor que 5% ou maior que 95% devido à possibilidade de crítico ou erro automático. Para o caso de salvaguardas, a equação é diferente:

$$HIT = \frac{\min(20, \max(0, CD - SG - 1))}{20} \quad (4.2)$$

Em que CD é a classe de dificuldade do ataque do atacante e SG é a salvaguarda relevante do alvo. A principal diferença entre os dois casos é que, no primeiro, a rolagem do d20 será aplicada ao bônus de ataque para tentar ultrapassar a classe de armadura do alvo, enquanto no segundo, a rolagem do d20 será aplicada à salvaguarda do alvo para tentar resistir à classe de dificuldade do atacante. Ou seja, uma rolagem maior no dado dá uma chance maior de acertar no primeiro caso, e uma chance menor de acertar no segundo. Isso é importante para discutir o conceito de vantagem e desvantagem, que modificam a chance de acertar. Com vantagem, para o caso da equação 4.1, a chance se torna:

$$HIT_A = 1 - (1 - HIT)^2 \quad (4.3)$$

Indicando que o ataque erra somente se ambas as rolagens errarem. Já com desvantagem, a chance se torna:

$$HIT_D = HIT^2 \quad (4.4)$$

Indicando que o ataque acerta somente se ambas as rolagens acertarem. Para o caso da equação 4.2, As equações são aplicadas ao contrário. Nesse caso, vantagem na rolagem significará uma chance menor de acertar, e desvantagem uma chance maior de acertar.

Para o dano em si, é importante saber que o valor médio rolado em um dado qualquer é a soma de todos os seus lados dividido por seu maior valor. Um d6, por exemplo, tem resultado médio  $1+2+3+4+5+6/6 = 3.5$ . Utilizando esse fato, é fácil calcular o dano médio de um ataque como:

$$Dano = QuantidadeDados * MediaDado + Modificador \quad (4.5)$$

Em que Dano é o dano médio esperado do ataque, QuantidadeDados é a quantidade de dados que serão rolados, MédiaDado é o resultado médio do dado, como descrito acima, e modificador é o bônus adicionado após rolar o dado. Por exemplo, um ataque cujo dano é  $2d6+3$  daria, em média, 10 pontos de dano. Se o ataque possui múltiplas rolagens de dano diferentes, é possível calcular cada uma individualmente e somá-las ao final.

Assim, o dano médio final de um ataque esperado é dado por:

$$DanoFinal = Dano * HIT \quad (4.6)$$

Em que DanoFinal é o dano dado considerando a chance de acertar e o dano médio do ataque quando acerta. No caso das salvaguardas, frequentemente quando o alvo obtém sucesso, o dano é apenas reduzido pela metade, e não nulificado. Para essas ações, a equação será:

$$DanoFinal = Dano * HIT + \frac{Dano}{2} * (1 - HIT) \quad (4.7)$$

O valor *DanoFinal* é utilizado para calcular a pontuação da ação em relação a seu alvo, como:

$$PontuacaoDano = DanoFinal * Modificadores \quad (4.8)$$

Em que *Modificadores* é uma coleção de aspectos que são determinados pelo usuário da ação e pelo alvo. Alguns exemplos incluem resistências, vulnerabilidades e imunidades a diferentes tipos de dano ou uma preferência do usuário a usar ações de certo tipo ou contra certos alvos, dependendo de sua definição de AI.

*PontuacaoDano*, então, pode ser comparado aos valores de outras ações e considerados contra outros alvos para encontrar a melhor ação possível contra os melhores alvos possíveis. Se a ação irá afetar múltiplos alvos, ou múltiplas ações serão realizadas em sequência, esses valores são somados para obter a pontuação final.

Para ações que recuperam pontos de vida para aliados, seu valor médio pode ser calculado como:

$$Recuperacao = QuantidadeDados * MediaDado + Modificador \quad (4.9)$$

Que é a mesma equação que o cálculo de dano. Um aspecto a ser considerado é que todo ator possui uma quantidade de pontos de vida máximo, e qualquer recuperação que leva a vida a um valor além desse máximo será desperdiçada. Assim, a pontuação de uma ação de recuperação de vida para um alvo pode ser definida como:

$$PontuacaoRecuperacao = \min(MHP - HP, Recuperacao) * Modificadores \quad (4.10)$$

Em que *MHP* é a quantidade máxima de pontos de vida do alvo, e *HP* é sua quantidade atual de pontos de vida.

Finalmente, para ações que causam condições, existem inúmeras condições diferentes com efeitos variados, sejam pequenas vantagens ou efeitos que quase certamente levarão à vitória no combate. Devido à complexidade das diferentes condições, foi definido que o valor de todas as condições será calculada da mesma maneira, como:

$$ValorCondicao = \frac{HP}{2} * HIT * Modificadores \quad (4.11)$$

Com a lógica que um alvo com mais pontos de vida tende a ser mais forte e se manter vivo durante mais turnos em combate, portanto, uma condição terá efeitos mais significativos quanto mais pontos de vida o alvo tiver. Isso é verdade ambas para condições positivas aplicadas em aliados quanto para condições negativas aplicadas em inimigos. É

importante notar que em D&D, uma criatura só pode ser afetada por uma condição com o mesmo nome, então definimos a pontuação final de ações que causam condições como:

$$PontuacaoCondicao = \begin{cases} 0, & \text{se já possui condição com mesmo nome.} \\ ValorCondicao, & \text{caso contrário} \end{cases}$$

Após calculadas as possíveis ações e definidas quais são as melhores, as ações devem ser executadas. Para isso, seguem o algoritmo 2

**Input:** *Acao*, a ação sendo executada e *Alvos*, uma listas de atores cuja ação pretende afetar

```

for Ator in Alvos do
  tentativa ← Falha;
  if Acao utiliza uma rolagem de ataque then
    if  $1d20 + \text{bônus de ataque de } Acao \geq \text{classe de armadura de } Ator$  then
      tentativa ← Sucesso;
    end
  else if Acao força uma salvaguarda then
    if  $1d20 + \text{salvaguarda de } Ator < \text{classe de dificuldade de } Acao$  then
      tentativa ← Sucesso;
    end
  else if Acao sempre funciona then
    tentativa ← Sucesso;
  end
  if Acao causa dano then
    if tentativa == Sucesso then
      Ator registra dano (tipicamente uma rolagem de dados no formato XdY+Z);
    else if Acao causa dano reduzido em falha then
      Ator registra dano reduzido (tipicamente uma rolagem de dados no formato XdY+Z);
    end
  else if Acao recupera pontos de vida then
    if tentativa == Sucesso then
      Ator registra recuperação de vida (tipicamente uma rolagem de dados no formato XdY+Z);
    end
  else if Acao aplica uma condição then
    if tentativa == Sucesso then
      Ator recebe uma condição, cujos efeitos são aplicados imediatamente.;
    end
  end
end

```

**Algorithm 2:** Algoritmo para a execução de uma ação contra um alvo

### 4.3 MÉTRICAS DE BALANCEAMENTO

Uma grande quantidade de simulações gera uma grande quantidade de informações brutas, mas que não são facilmente compreendidas devido a seu volume. Para obter uma análise útil dos resultados dos combates, são calculadas métricas de balanceamento, que permitem entender não só a dificuldade de um combate, mas também detalhes sobre o como será esse encontro. Com as métricas, é possível observar se um combate levará uma grande quantidade de rodadas ou será mais rápido, se necessitará de um gasto grande de recursos para vitória, se será constantemente vantajoso para um lado ou se cada lado constantemente tentará conseguir a vantagem, entre outras medidas úteis.

A seguir, serão descritas métricas úteis para o DM. Para algumas dessas métricas, é utilizado o conceito de vantagem, indicando que um lado atualmente está demonstrando uma performance melhor que o outro. No contexto de um combate em D&D, a vantagem pode ser calculada no final de cada rodada como a diferença entre as porcentagens restantes de pontos de vidas totais de cada time, com o time com a maior porcentagem atualmente possuindo a vantagem.

- Porcentagem de vitórias: A porcentagem de simulações em que cada lado foi vitorioso. Em D&D, é esperado que esse valor fique próximo de 100% a favor dos personagens dos jogadores.
- Duração média: A quantidade de rodadas média esperada para o término do combate. É comum um valor de aproximadamente três rodadas, mas encontros mais curtos ou mais longos podem ser interessantes, a depender dos propósitos do DM.
- Mortes totais: A quantidade de vezes que cada ator morreu ao longo das simulações. Essa informação pode ajudar o DM a planejar melhor suas estratégias, sabendo quais inimigos ou personagens são mais vulneráveis ou mais difíceis de serem derrotados.
- Porcentagem de Pontos de vida restantes: A porcentagem média de pontos de vida restantes de cada time em simulações onde esse time foi vitorioso. Um valor menor indica que o combate foi mais perigoso, e necessitou de mais recursos para vitória.
- Incerteza média: Uma porcentagem indicando o quão frequentemente a vantagem trocou de lado. Um valor alto indica que o combate foi próximo, frequentemente trocando qual lado possuía a vantagem, enquanto um valor mais baixo indica que um dos dois times estava constantemente favorecido.
- Permanência média: Um valor de 0 a 1 indicando, para cada time, por quanto tempo conseguiu manter a vantagem no combate, sem perdê-la. Um valor próximo de 1 indica que o time frequentemente conseguiu manter a vantagem durante a maioria do combate, enquanto um valor próximo de 0 indica que o time raramente conseguiu a vantagem, ou que os times trocaram de vantagem frequentemente.

- Vantagem decisiva: O valor de vantagem necessário para garantir a vitória de um dos dois lados do combate. Como D&D é um jogo influenciado pela sorte, certamente é sempre possível que um dos lados consiga vencer, mesmo quando o outro lado possuir uma grande vantagem, mas quanto maior a vantagem, mais improvável disso acontecer. Esse valor, calculado separadamente para cada time, indica a maior vantagem encontrada nas simulações em que o time oponente foi capaz de vencer, ou seja, qualquer vantagem acima desse valor não dá chance de vitória ao time oponente.

#### 4.4 ESCOPO DO PROJETO

O simulador criado deve ser capaz de realizar a simulação de um encontro de combate entre dois grupos de atores, utilizando as regras de D&D. Deve ser possível criar atores customizados, para representar os personagens dos jogadores e os inimigos que encontram, além de ações e condições associadas a esses atores. Finalmente, a simulação deve devolver ao usuário métricas úteis para análise e balanceamento dos encontros.

D&D é um jogo extremamente complexo e possui múltiplas características que o deixam mais difícil de simular, que não serão adaptadas diretamente para o simulador. Notavelmente, não será implementado posicionamento e movimentação, e não serão implementadas reações. Esses são aspectos importantes do jogo, mas outros elementos serão ajustados para ajudar a substituí-los.

Além disso, será possível simular múltiplos combates em sequência, considerando um time principal que encontra grupos de inimigos sucessivamente, com possibilidade de recuperação limitada de vida e recursos entre combates. Em uma campanha real de D&D, é comum ocorrerem múltiplos combates seguidos, drenando os recursos dos jogadores ao longo do tempo, tipicamente em masmorras em que é impossível obter o tempo necessário para recuperar todos os recursos.

#### 4.5 REQUISITOS FUNCIONAIS

- O simulador deverá ser capaz de simular um combate entre dois grupos de atores.
  - O simulador usará uma estratégia de escolha de ação especificada para cada ator entre opções simples.
  - O simulador deve simular as mecânicas da maneira como funcionam em D&D para obter uma simulação mais próxima da realidade possível.
  - O simulador deve poder realizar uma quantidade arbitraria de simulações definida pelo usuário
- A interface deve ser capaz de criar e editar atores e suas ações e condições, e chamar uma simulação de combate com os atores criados.

- Os atores, além de suas ações e condições, devem ser armazenados localmente.
- A simulação deve ser rodada ao vivo, devolvendo os resultados de múltiplas simulações de combate e uma reportagem detalhada de cada combate.

#### 4.6 REQUISITOS NÃO-FUNCIONAIS

Nesta parte usamos a classificação de (SOMMERVILLE, 2015) para requisitos não funcionais.

- Requisitos do produto
  - Requisitos de usabilidade
    - \* Um DM experiente deve ser capaz de, após duas horas de experimentação e familiarização com a ferramenta, conseguir criar, editar e simular um combate entre um grupo de personagens feitos por jogadores e um ou mais grupos de monstros, com um máximo de dois erros de uso por simulação em média.
    - \* Um usuário familiar com a ferramenta deve ser capaz de exportar um JSON de uma ator, ação ou condição. Outro usuário deve em seguida ser capaz de importar esse JSON para utilização e modificação.
    - \* Um usuário deve ser capaz de acessar diretamente os dados abertos dos atores, ações e condições, e adicionar ou remover como desejar.
  - Requisitos de eficiência
    - \* Requisitos de desempenho
      - Uma simulação de 1000 combates entre dois atores não deve exceder dois segundos. Para cada ator além desses dois, o limite aumenta em mais dois segundos. Para múltiplos combates em sequencia, o limite deve ser calculado para cada combate e somados.
    - \* Requisitos de espaço
      - A ferramenta em si não deve exceder 50 MB após compilada. Os arquivos JSON armazenando atores, ações e condições, cuja quantidade é indeterminada e definida pelas necessidades do usuário, não devem exceder 50 KB cada.
- Requisitos da organização
  - Requisitos de prazo
    - \* O projeto deve ser finalizado até a data 30/07/2024.
- Requisitos externos

– Requisitos legais

- \* Em sua versão de lançamento, não devem ser inclusos atores, ações ou condições fora aqueles disponibilizados pela *Wizards of the Coast* em seu *Systems Reference Document* (SRD) de D&D 5e.

## 4.7 AVALIAÇÕES

Para garantir que a ferramenta desenvolvida estará servindo a seus propósitos de conseguir simular encontros de combate com acurácia às regras de D&D e com isso auxiliar DMs na criação de encontros de combate, serão realizadas avaliações quantitativas e qualitativas.

### 4.7.1 Avaliações Quantitativas

O *Dungeon Master's Guide* inclui regras simplificadas para balanceamento de encontros de combate considerando apenas o nível dos personagens, o CR dos inimigos e a quantidade de atores de cada lado. Para garantir que os resultados obtidos pelo simulador são consistentes com o que se espera de encontros de combate reais, serão realizados diversos testes compostos de combates reais, em níveis diferentes e contra inimigos diferentes, buscando comparar os resultados obtidos para a dificuldade através do simulador e das regras oficiais.

### 4.7.2 Avaliações Qualitativas

A avaliação qualitativa buscará obter as opiniões sobre a usabilidade e acurácia da ferramenta do ponto de vista dos DMs, através de entrevistas e questionários, com uma combinação de perguntas abertas e de afirmações feitas utilizando escalas de Likert.

#### 4.7.2.1 Escalas de Likert

Escalas de Likert serão utilizadas para a avaliação qualitativa do simulador. Múltiplas afirmações sobre diferentes aspectos do simulador são feitas, incluindo sua facilidade de uso e a percepção da acurácia dos combates, e o usuário classifica cada afirmação em uma escala de 1 (discordo muito) a 5 (concordo totalmente).



## 5 IMPLEMENTAÇÃO

Para podermos analisar os resultados de múltiplos combates de D&D, e permitir que DMs sejam capazes de fazer o mesmo, com seus combates e personagens específicos, foi criado um simulador de combates em D&D, que permite ao usuário fazer o design de atores, múltiplos tipos diferentes de ações, e diferentes tipos de condições aplicadas aos atores, e em seguida simular o resultado de múltiplos combates entre os dois. A seguir será descrita a implementação desse simulador.

### 5.1 FERRAMENTAS UTILIZADAS

O simulador foi criado usando Python, pois é uma linguagem simples de entender e utilizar e possui grande quantidade de bibliotecas úteis. Para a interface, originalmente foi planejado o uso de PyQt5, um grupo de bibliotecas capaz de criar interfaces de alto nível, porém foi decidido que era mais complexa e avançada que o necessário, então foi utilizada a biblioteca PySimpleGUI, que efetivamente faz muito do mesmo e é mais simples de utilizar.

A aplicação foi feita para ser utilizada de forma inteiramente local, com os dados de atores, ações e condições sendo armazenados junto ao programa em formato JSON. Isso foi feito para obter algumas vantagens, ambas pequenas e grandes: Não é necessário acesso à internet para utilizar, não foi necessário desenvolver servidores para tratar os dados, e é fácil baixar conteúdo criado por outros fazendo apenas download de seus JSONs.

### 5.2 EVOLUÇÃO DO SIMULADOR

Durante o desenvolvimento, o simulador passou por um total de 7 versões, cada uma delas adicionando mais conteúdo em relação à versão anterior. O simulador e suas versões podem ser conferidos em <https://github.com/Marcelo-Drummond-Fonseca/DnD-Simulator>

#### 5.2.1 Versão 1

Essa versão incluiu o mínimo necessário para simular combate entre dois atores simples, sem nenhuma propriedade especial e sem nenhuma ação além de um ataque básico. Foi criada uma função simples para rolar dados, que eventualmente foi adaptada para considerar detalhes mais específicos como vantagem, além de uma classe “*Creature*” e uma classe “*Action*” simples, que incluíam apenas o mínimo necessário para conseguir realizar o combate seguindo as regras do jogo.

Ao iniciar a simulação, ambos atores rolam iniciativa, definindo qual deles agirá primeiro. Em seguida, alternam seus ataques um contra o outro, rolando dados para conferir se acertam e definir o dano dado, com o primeiro cujos pontos de vida chega a zero primeiro sendo o perdedor.

### 5.2.2 Versão 2

Essa versão definiu a existência de diferentes tipos de dano. D&D inclui um total de 13 tipos de dano: Ácido, contundente, cortante, elétrico, energético, gélido, ígneo, necrótico, perfurante, psíquico, radiante, trovejante e venenoso. Todo ator pode ser neutro, resistente, imune ou vulnerável a cada um desses tipos de dano, com atores resistentes tomando metade do dano, vulneráveis tomando o dobro do dano, e imunes ignorando o dano completamente.

Foi adicionado aos ataques e os atores um tipo de dano e se o ator possui alguma resistência, imunidade ou vulnerabilidade específica, para que funcione de forma semelhante a como é em D&D. Outros jogos famosos, como *Pathfinder*, utilizam uma versão diferente de resistência e vulnerabilidade, em que ao invés de cortar na metade ou dobrar, cada criatura possui um valor fixo que é reduzido ou adicionado ao dano tomado (por exemplo, uma criatura com resistência a fogo 10 tomando 15 pontos de dano de fogo perderia apenas 5 pontos de vida). As versões iniciais do simulador também possuem a estrutura para utilizar esse tipo de tratamento, mas versões futuras não desenvolveram com seu uso, devido ao foco na quinta edição de D&D.

### 5.2.3 Versão 3

O foco dessa versão foi em adicionar ações mais complexas e tudo relacionado a elas. Primeiramente, muitas ações, como por exemplo magias, tem algum custo associado a elas, fazendo o ator utilizando a ação perder algum recurso. O custo de cada ação e os recursos disponíveis ao ator mudam para cada ação e cada ator, assim devem ser definidos pelo usuário, enquanto o simulador trata apenas de manter em mente quais recursos foram gastos e quais ações são possíveis de serem realizadas com os recursos restantes para o ator.

Além disso, foi adicionada a possibilidade de um ator poder utilizar um “combo” de ações. As classes de personagens com foco em dano de armas e a maioria dos monstros de nível alto possuem habilidades como “*Extra Attack*” e “*Multiattack*”, que permitem que realizem múltiplas ações em um turno.

Mas o maior foco dessa versão foi na criação da estrutura das ações. Cada ação tem duas etapas: uma etapa conferindo se a ação funciona, definida como tentativa, e uma etapa aplicando o efeito da ação, definida como efeito. Cada etapa possui três tipos, levando a um total de nove possíveis combinações:

- Tentativas

- Ataque: Nessa tentativa, o usuário da ação rola 1d20, adicionando algum modificador definido pela ação e seus atributos. Se esse valor for maior ou igual à classe de armadura do alvo, a tentativa é um sucesso. Adicionalmente, se o dado rolar um 20, é obtido um sucesso crítico.
- Salvaguarda: Nessa tentativa, o alvo da ação rola 1d20 e adiciona uma de suas salvaguardas, a salvaguarda específica sendo determinada pela ação. Se o valor for maior ou igual a um valor determinado na ação, o efeito no alvo é reduzido ou pode até ser nulificado.
- Automático: Essa tentativa sempre obtém sucesso. É o caso, por exemplo, de grande parte dos efeitos que afetam o próprio usuário ou seus aliados.

- Efeitos

- Dano: Esse efeito faz o alvo perder pontos de vida, com a quantidade sendo definida por uma rolagem no dado além de quaisquer outros efeitos e modificadores sendo aplicados no atacante e/ou no alvo.
- Recuperação de Vida: Esse efeito, usado quase exclusivamente nos aliados, faz o alvo recuperar pontos de vida para ter uma chance menor de morrer.
- Aplicar Condição: Esse efeito aplica uma condição no alvo. Esse efeito não foi implementado nessa versão, e será descrito em detalhes na seção 5.2.4.

É também possível que uma ação tenha múltiplos efeitos. Por exemplo, a magia *Ray of Sickness* faz uma tentativa de ataque com efeito de dano, e se o ataque for um sucesso, o alvo também precisa fazer uma salvaguarda para tentar evitar receber a condição "envenenado". Para tratar casos desse tipo, foi adicionada ao simulador a possibilidade de ações terem uma ação seguinte ao obter um sucesso.

#### 5.2.4 Versão 4

O foco dessa versão foi adicionar condições e seus efeitos. Em D&D, “condições” tipicamente se refere a um conjunto de efeitos que podem ser aplicados a um ator, dando a ele algum tipo de efeito negativo. As condições em D&D oficial são agarrado, amedrontado, atordoado, caído, cego, enfeitado, envenenado, impedido, incapacitado, inconsciente, invisível, paralisado, petrificado e surdo.

No entanto, existem múltiplos efeitos que não são considerados condições em D&D, mas que afetam o personagem de alguma forma semelhante. a habilidade de bárbaros “*Reckless Attack*”, por exemplo, da vantagem em ataques que eles fazem, mas vantagem em todos os ataques feitos contra eles. Atores diferentes tem acesso a inúmeras habilidades capazes de fazer combinações de diversos efeitos. Assim, no contexto do simulador,

condições foram abstraídas, sendo uma coleção nomeada de diferentes efeitos aplicados a um ator, e removidos quando a condição acaba. Assim, abrangem ambos os efeitos de condições de D&D quanto múltiplas outras habilidades de diferentes atores.

### 5.2.5 Versão 5

Com a conclusão de condições, foi dada uma pausa na parte da simulação em si para ser feita a construção da interface para o usuário. Como o objetivo é que um DM possa utilizar a ferramenta para balancear seus encontros de combate, é importante que ele possa criar e editar criaturas e encontros com facilidade.

A interface foi separada em 4 seções, uma para criaturas, uma para ações, uma para condições e uma para o simulador. As três primeiras permitem realizar a criação e edição de seus respectivos elementos, que são salvos localmente na forma de JSONs. a aba representando o simulador permite a adição de criaturas para dois times diferentes, que entram em combate um contra o outro. Após o combate, os resultados são devolvidos ao usuário.

### 5.2.6 Versão 6

Anteriormente, as ações tomadas por cada ator eram decididas aleatoriamente, contanto que tivessem os recursos para realizá-la. Nesse ponto, foi incluído um grupo de regras para ajudar os atores a tomarem ações de forma mais eficaz.

Primeiramente, são definidas as ações possíveis de serem realizadas. após filtradas, o ator olha para os combos que sobraram. Para cada combo, irá gerar uma lista de valores que representam o valor que esse combo possui contra cada possível alvo. Esse valor é calculado como a média de dano ou recuperação de vida que o combo dará, com dois modificadores, um dependendo do tipo de ação e um dependendo do tipo do alvo, em relação ao tipo do ator.

Os atores foram separados em quatro tipos, “*Tank*”, “*Damage*”, “*Support*” e “*Neutral*”. Os três primeiros representam as funções clássicas de um grupo de personagens em RPG, enquanto o último indica nenhuma preferência em termos de ações. Tanques gostam de por condições positivas em si próprios e dar dano em outros tanques, e são os principais alvos de ataques. Personagens focados em dano tentam derrotar os alvos mais fracos enquanto sendo atacado menos que os tanques. Finalmente, personagens de suporte dão preferência a recuperar a vida de seus aliados e oferecer condições positivas a eles, especialmente tanques.

Tendo os valores de cada combo em mãos, o ator decide, entre os combos com uma pontuação acima da média, qual deles será o utilizado aleatoriamente, com peso igual à pontuação.

### 5.2.7 Versão 7

Essa versão, que foi continuamente atualizada durante o resto do projeto, tem três funções. Primeiramente, foram realizadas múltiplas limpezas da interface, para ser mais intuitiva para usuários inexperientes. Além disso, nessa versão foi implementada a possibilidade de serem realizados múltiplos combates em sequência. Finalmente, a outra função principal dessa versão é encontrar e consertar qualquer tipo de erro encontrado durante execuções e testes da ferramenta.

## 5.3 ARQUITETURA FINAL

### 5.3.1 Diagrama de Classes

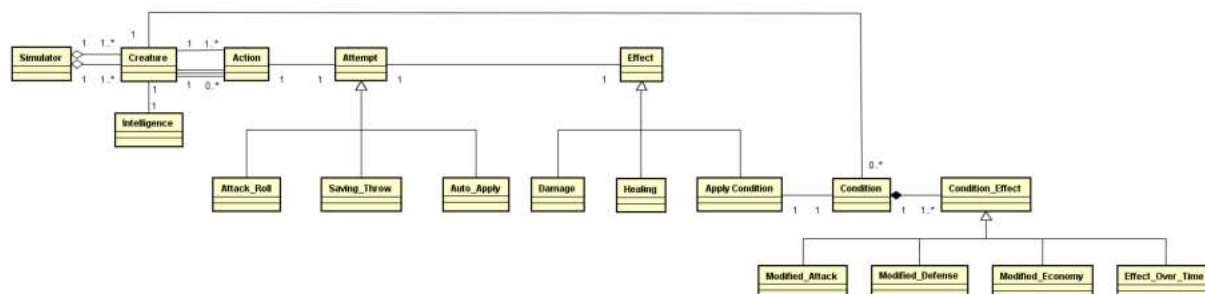


Figura 2 – O diagrama de classes do aplicativo.

A figura 2 mostra o diagrama de classes utilizado para a simulação. Há apenas uma instância da classe *Simulator*, seguindo o padrão *Singleton*. Essa classe é responsável por armazenar e calcular diversos dados sobre a simulação que serão devolvidos ao usuário, como quantidade de turnos, porcentagem de pontos de vida restantes de cada time em cada turno, e mortes em combate. Ela armazena dois times de criaturas, ambos compostos de uma ou mais criaturas inicialmente, e após definida a ordem de iniciativa, avisa em ordem quando que cada ator deve agir.

Os atores são o ponto focal da simulação. Além de armazenar todos os atributos básicos necessários para o jogo de D&D, como pontos de vida, classe de armadura, salvaguardas, entre outros, a classe *Creature* também monitora suas vantagens e desvantagens, a quantidade de recursos de cada tipo que possui, se está concentrando em algum efeito, etc.

Cada ator possui uma instância de *Intelligence*, com valores de prioridade para ações e alvos diferentes, que podem ser escolhidos entre os quatro valores padrões. Essa classe é utilizada para definir qual ação que o ator tomará contra qual alvo. Os atores também podem possuir condições permanentes, como é o caso de *Rogues* com *Evasion*, mas isso é relativamente raro. Finalmente, um ator possui quatro listas de ações, referentes aos

diferentes tipos de ações incluindo ações normais, ações bônus, ações "gratuitas" e ações de sequência. É necessário que um ator possua ao menos uma ação normal para que consiga funcionar em combate, todos os outros tipos são opcionais.

Como descrito anteriormente, as ações são divididas em pedaços, com a classe *Action* incluindo as características comuns a todas as ações. Quando a função *act* de uma ação é chamada, recebendo a criatura que usou a ação e o alvo da ação como argumentos, ela em sequência chamará a mesma função em sua classe *Attempt* correspondente. A classe *Attempt* realizará o método definido por sua instância para definir se a ação obteve sucesso ou não e em seguida chamará a função na classe *Effect*, que aplicará o efeito correspondente no alvo, dependendo dos resultados da tentativa.

Finalmente, as condições possuem uma classe principal *Condition* que representa a condição em si, incluindo todos os valores comuns a todas as condições, incluindo sua duração e em que momento essa duração diminui (tipicamente, no começo ou fim do turno do ator que aplicou a condição ou do alvo da condição). A classe possui uma lista de instâncias da classe *Condition\_Effect*, que representa a maneira em que a condição afeta o ator ao qual está aplicada. Quando a condição é aplicada ao ator, imediatamente todos os efeitos da condição são aplicados a seus atributos, e quando a condição é removida, esses efeitos são revertidos.

## 5.4 INTERFACE/ COMO FUNCIONA

A interface foi criada para que DMs possam utilizar a ferramenta para realizar suas simulações, utilizando criaturas originais ou já prontas de livros, e os personagens de seus jogadores, que vão evoluindo ao longo do tempo.

A interface possui cinco abas principais: *Simulator*, *Creature*, *Action*, *Condition* e *Resources*. A primeira permite que o usuário planeje e prepare um encontro de combate ou uma sequência de encontros de combate, enquanto as outras permitem que diversos atributos dos atores sejam criados e editados. As subseções a seguir serão nomeadas tal qual no aplicativo.

As abas *Creature*, *Action* e *Condition* possuem alguns aspectos em comum. Todas possuem uma lista na esquerda com todos os editáveis correspondentes, além de uma barra de pesquisa para facilitar a busca e um botão “*Create New*”, que cria uma nova instância vazia do editável. Além disso, abaixo da área de edição à direita, há um botão para salvar todas as alterações feitas. Finalmente, cada aba é separada em telas mais específicas, permitindo a edição em partes menores.

### 5.4.1 Simulator

Na aba do simulador, é possível fazer times de atores diferentes lutarem entre si. Há espaço para serem definidos quatro times diferentes, que podem incluir qualquer combi-

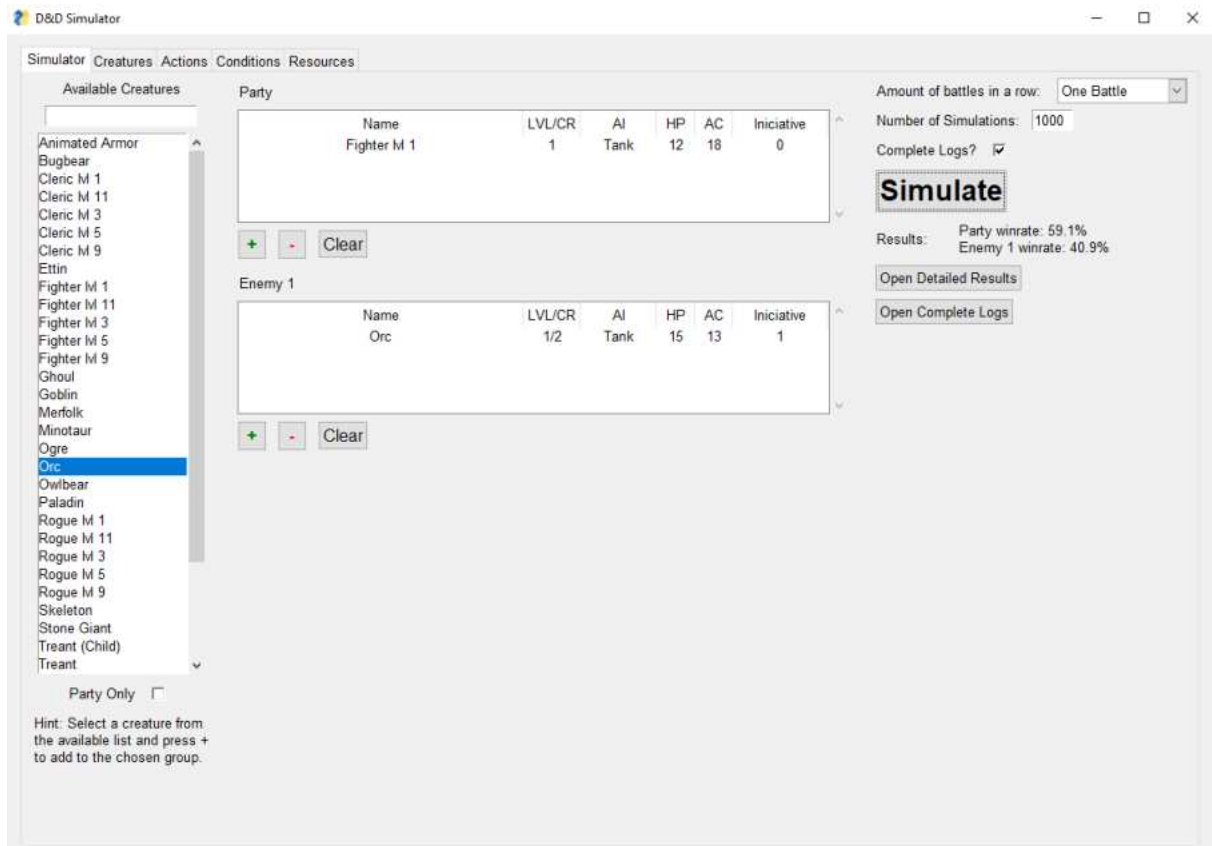


Figura 3 – Tela exemplificando uso do simulador para uma única luta.

nação de atores, incluindo múltiplas instâncias do mesmo ator.

Há dois casos diferentes para o uso do simulador. No primeiro caso, é preenchido o primeiro e o segundo time, e feita uma simulação de combate entre os dois. Após ser simulado múltiplas vezes, o simulador devolverá algumas métricas úteis, incluindo a porcentagem de vitória de cada lado, a quantidade média de pontos de vida restante do time vencedor, a quantidade de vezes que cada personagem morreu, entre outras. Essas métricas estarão disponíveis em um arquivo “*Results*”. Cada simulação também é salva para um arquivo “*Battle Log*”, que relata o passo-a-passo de cada um dos combates, caso seja de interesse ver mais detalhadamente os acontecimentos de cada. Ambos arquivos podem ser acessados pelos botões que aparecem após realizada a simulação.

No segundo caso, em que mais de dois times são preenchidos, será realizado o modo massmorra, em que o time 1 terá que enfrentar os times 2, 3 e 4 em ordem. Isso simula um caso mais real de se acontecer em campanhas de D&D, em que frequentemente ocorrem múltiplos combates seguidos. Pode ser definido se haverá algum descanso ou não entre cada combate, podendo ser imediato (indicando que o time seguinte aparece imediatamente depois da derrota do time anterior), sem descanso (indicando que houve tempo suficiente para as condições afetando os atores acabarem, mas não para tomarem um descanso) ou descanso curto (indicando que os atores puderam recuperar seus recursos de

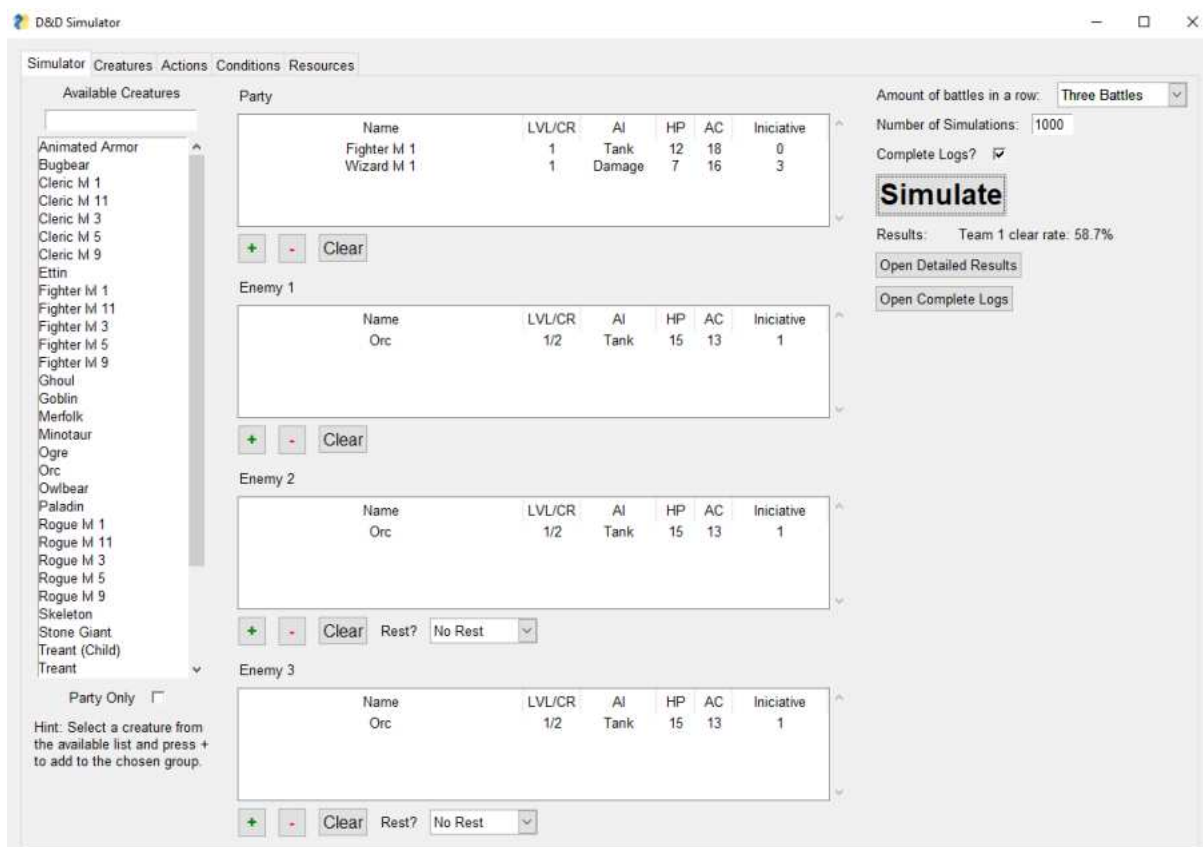


Figura 4 – Tela exemplificando uso do simulador para lutas em sequencia.

descanso curto, e parcialmente recuperar seus pontos de vida). Após simulada, o simulador apresentará métricas semelhantes ao do primeiro caso, mas dessa vez mais voltadas à performance do time 1. Novamente, tudo será visível detalhadamente na *Battle Log*.

#### 5.4.1.1 Results e Battle Log

Os resultados das batalhas são salvos em arquivos de texto comuns, que podem ser abertos diretamente pelos botões como descrito previamente.

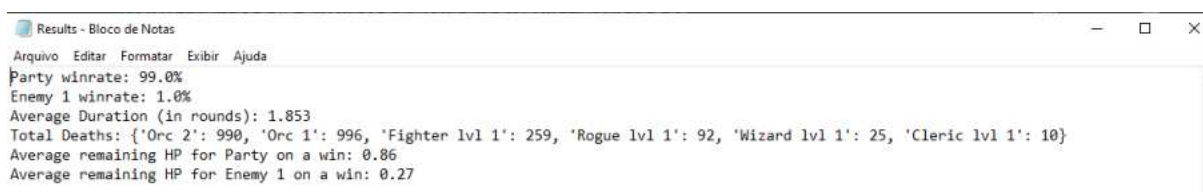


Figura 5 – Arquivo *Results* com as métricas simples obtidas pela simulação.

O arquivo *Results* inclui as métricas mais simples, que podem ser rapidamente vistas e deixam claro para o DM o que pode esperar do combate de forma geral.

Já o arquivo *Detailed Logs* inclui, de forma detalhada, absolutamente tudo que acontece em cada combate: Cada rolagem, cada decisão, cada ação e cada condição. Isso



```

Battle Log - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
2024-08-15 17:37:27,857 - Rolou 8 no dado e tem modifier 1
2024-08-15 17:37:27,857 - Cleric lvl 1 rolou 9 de iniciativa
2024-08-15 17:37:27,857 - Rolou 20 no dado e tem modifier 0
2024-08-15 17:37:27,857 - Fighter lvl 1 rolou 20 de iniciativa
2024-08-15 17:37:27,857 - Rolou 17 no dado e tem modifier 3
2024-08-15 17:37:27,857 - Rogue lvl 1 rolou 20 de iniciativa
2024-08-15 17:37:27,857 - Rolou 11 no dado e tem modifier 3
2024-08-15 17:37:27,857 - Wizard lvl 1 rolou 14 de iniciativa
2024-08-15 17:37:27,857 - Rolou 6 no dado e tem modifier 1
2024-08-15 17:37:27,857 - Orc rolou 7 de iniciativa
2024-08-15 17:37:27,857 - Turno de: Fighter lvl 1
2024-08-15 17:37:27,857 - Fighter lvl 1 usa Longsword contra ['Orc']
2024-08-15 17:37:27,857 - Rolou 20 no dado e tem modifier 5
2024-08-15 17:37:27,857 - Ataque de 25 acerta AC de 13
2024-08-15 17:37:27,857 - Critico!
2024-08-15 17:37:27,858 - Rolou [5, 6] no dado e tem modifier 5
2024-08-15 17:37:27,858 - Orc toma 16 de dano. está agora com -1 de vida
2024-08-15 17:37:27,858 - Orc morreu
2024-08-15 17:37:27,858 - Team 1 Wins!
2024-08-15 17:37:27,858 - Mortes: ['Orc']
2024-08-15 17:37:27,858 - Incerteza: 0.0
2024-08-15 17:37:27,858 - Duração: 1 rounds
2024-08-15 17:37:27,858 - A maior permanência foi de 1 rounds pelo time 1.
Permanência de 100.0%
2024-08-15 17:37:27,858 - A maior vantagem foi de 0
2024-08-15 17:37:27,858 - A porcentagem de HP restante do time vencedor foi 100.0%

2024-08-15 17:37:27,858 - Rolou 16 no dado e tem modifier 1
2024-08-15 17:37:27,858 - Cleric lvl 1 rolou 17 de iniciativa
2024-08-15 17:37:27,858 - Rolou 3 no dado e tem modifier 0
2024-08-15 17:37:27,858 - Fighter lvl 1 rolou 3 de iniciativa
2024-08-15 17:37:27,858 - Rolou 12 no dado e tem modifier 3
2024-08-15 17:37:27,858 - Rogue lvl 1 rolou 15 de iniciativa
2024-08-15 17:37:27,858 - Rolou 5 no dado e tem modifier 3
2024-08-15 17:37:27,858 - Wizard lvl 1 rolou 8 de iniciativa
2024-08-15 17:37:27,858 - Rolou 2 no dado e tem modifier 1
2024-08-15 17:37:27,858 - Orc rolou 3 de iniciativa
2024-08-15 17:37:27,858 - Turno de: Cleric lvl 1
2024-08-15 17:37:27,858 - Cleric lvl 1 usa Bless contra ['Fighter lvl 1', 'Cleric lvl 1', 'Wizard lvl 1']
2024-08-15 17:37:27,858 - Fighter lvl 1 está afetado pela condição: Blessed
2024-08-15 17:37:27,858 - Cleric lvl 1 está afetado pela condição: Blessed
2024-08-15 17:37:27,858 - Wizard lvl 1 está afetado pela condição: Blessed
2024-08-15 17:37:27,858 - Turno de: Rogue lvl 1
2024-08-15 17:37:27,858 - Rolou [1] no dado e tem modifier 0
2024-08-15 17:37:27,859 - Rogue lvl 1 Recarrega seu Sneak Attack
2024-08-15 17:37:27,859 - Rogue lvl 1 usa Rapier (Sneak Attack) contra ['Orc']
2024-08-15 17:37:27,859 - Rolou 18 no dado e tem modifier 5
2024-08-15 17:37:27,859 - Ataque de 23 acerta AC de 13
2024-08-15 17:37:27,859 - Rolou 13 no dado e tem modifier 3

```

Figura 6 – Arquivo *Battle Log* descrevendo cada batalha passo a passo incluindo suas métricas.

permite que o usuário realmente observe o desenrolamento dos combates, conseguindo observar o que funcionou, o que não funcionou, que tipo de sorte pode ser necessária, como foi a performance de atores específicos, entre muitas outras observações úteis. Após cada combate, são também incluídas as métricas calculadas para esse combate, incluindo as métricas mais avançadas relacionadas a vantagem, e no final, as métricas totais da simulação.

As métricas devolvidas incluem a porcentagem de vitória de cada lado, a duração média em rodadas, uma listagem da quantidade de mortes totais que ocorreram ao longo das simulações, a porcentagem de pontos de vida restantes para cada time nos combates em que foram vitoriosos, a incerteza e permanência médias, e a vantagem decisiva para cada lado. A vantagem é um valor entre -1 e 1, igual à subtração da porcentagem de pontos de vida restantes do time 1 (A *Party*) pela porcentagem de pontos de vida restantes do time 2, os inimigos.

```

2024-08-15 17:37:29,092 - Party winrate: 100.0%
Enemy 1 winrate: 0.0%
2024-08-15 17:37:29,092 - Duração média: 1.123 rounds
2024-08-15 17:37:29,092 - mortes (totais): {'Orc': 1000, 'Fighter lvl 1': 89, 'Rogue lvl 1': 8}
2024-08-15 17:37:29,092 - Incerteza média: 0.001
2024-08-15 17:37:29,092 - Permanencia média do time 1: 0.998
2024-08-15 17:37:29,092 - Permanencia média do time 2: 0.001
2024-08-15 17:37:29,092 - Vantagem Decisiva para time 1: 0.025000000000000022
2024-08-15 17:37:29,092 - Vantagem Decisiva para time 2: N/A
2024-08-15 17:37:29,092 - Porcentagem de HP restante médio para time 1: 0.95109375
2024-08-15 17:37:29,092 - Porcentagem de HP restante médio para time 2: 0
2024-08-15 17:37:29,092 - Tempo de execução: 1.234999418258667

```

Figura 7 – Métricas totais devolvidas no final do arquivo *Battle Log*

## 5.4.2 Creature

Essa aba permite que sejam criados e editados os diversos aspectos dos atores. Possui uma tela principal, e cinco telas de edição de aspectos mais específicos que aparecem como *popups*: *Resistances*, *Actions*, *Conditions*, *Resources* e *Combos*.

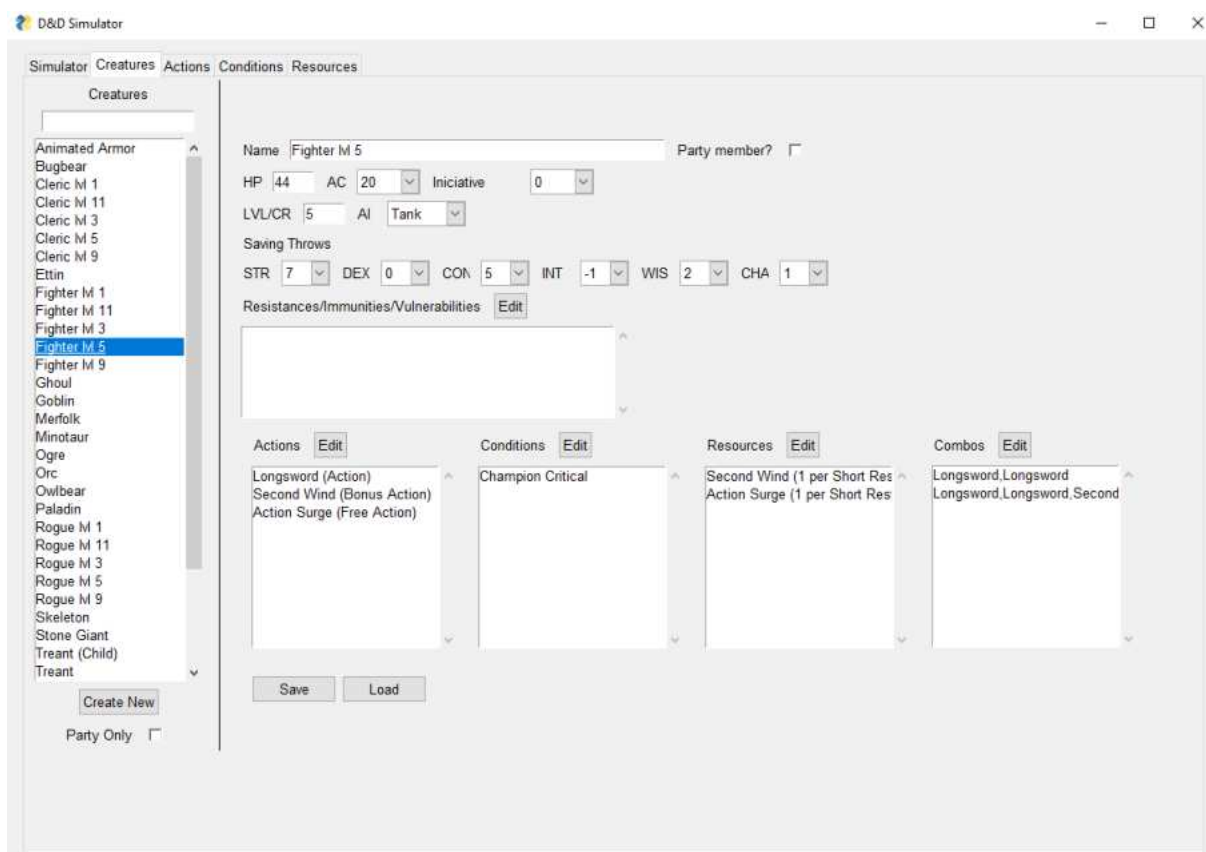


Figura 8 – Tela de edição dos atributos dos atores.

A tela principal engloba todas as características básicas necessárias para os atores funcionarem em um combate de D&D, incluindo seu nome, pontos de vida, classe de armadura, iniciativa e salvaguardas.



Figura 9 – Enter Caption

Além dessas características, aqui também pode ser definido o tipo de inteligência que o ator utilizará em combate. Tipicamente em RPGs existem três funções principais a serem preenchidas: O tanque, responsável por ser o personagem da linha de frente que é equipado para aguentar muito dano e evitar que os aliados mais fracos morram; o DPS (*Damage per Second*), cuja função é dar o máximo de dano possível para eliminar os oponentes na menor quantidade de tempo; e o suporte, responsável por recuperar a vida dos aliados para evitar que morram e usar habilidades para fortalecer seus aliados ou deixar os inimigos mais fracos. Esses três foram adaptados para os AIs selecionáveis “*Tank*”, “*Damage*” e “*Support*”, que possuem pesos diferentes para ações e alvos que correspondem a seu tipo de AI, além do “*Neutral*” que não possui nenhum peso a mais em qualquer direção.

Também é possível marcar um ator como sendo parte de uma *party*. Efetivamente, isso permite que o usuário encontre esses atores mais facilmente nas abas de simulação e de edição de criaturas. É útil para marcar personagens que serão utilizados múltiplas vezes, como os personagens de outros jogadores.

#### 5.4.2.1 Resistances

Diferentes atores podem ser afetados por tipos diferentes de dano de maneira específica. Os casos mais comuns são atores possuírem resistência ou imunidade a certos tipos de dano, fazendo com que recebam metade do dano, ou completamente nulifiquem o ataque. É possível também serem vulneráveis a certos tipos de dano, dobrando o dano que recebem. Nessa tela, é possível definir quais tipos de dano sofrerão essas diferenças.

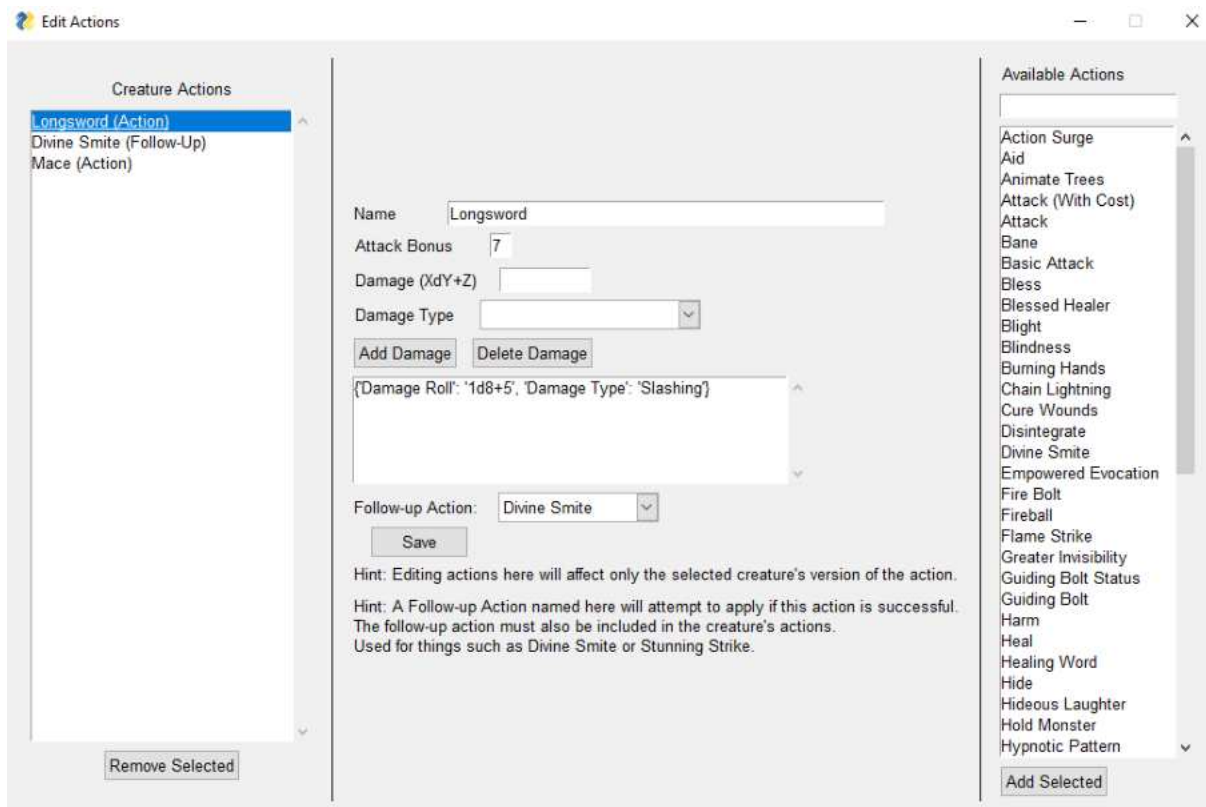


Figura 10 – Tela de edição das ações dos atores.

#### 5.4.2.2 Actions

Existem múltiplas ações que são comuns a diferentes atores, apenas com pequenas alterações. Quase todo ator possui alguma variação de um ataque simples, e vários tipos de magia são compartilhados entre diferentes magos. No entanto, devido a diferenças em atributos e habilidades dos atores, podem haver diferenças em aspectos como quantidade de dano, tipo de dano, CD, bônus de ataque, entre outros. Nessa tela, é possível editar as características das versões específicas das ações que o ator usa.

Além disso, nessa tela também é possível definir *Follow-up Actions*. Existem casos de ações no jogo que tem múltiplos efeitos diferentes, ou que podem ser feitas em sequencia a outras ações. No simulador, ao definir que uma ação possui uma *follow-up action*, isso quer dizer que, caso a ação tenha sucesso, o ator tentará realizar a *follow-up action* em seguida. Isso é útil para diversos tipos de atores como paladinos, que podem utilizar um *Divine Smite* após acertar um ataque, Monges, que podem utilizar *Stunning Strike* após acertar um ataque, diversos tipos de magias e ataques que, além de dar dano, causam efeitos negativos, como os chifres de um minotauro, que quando acertam um oponente, causam dano e o forçam a realizar uma salvaguarda contra a condição caído.

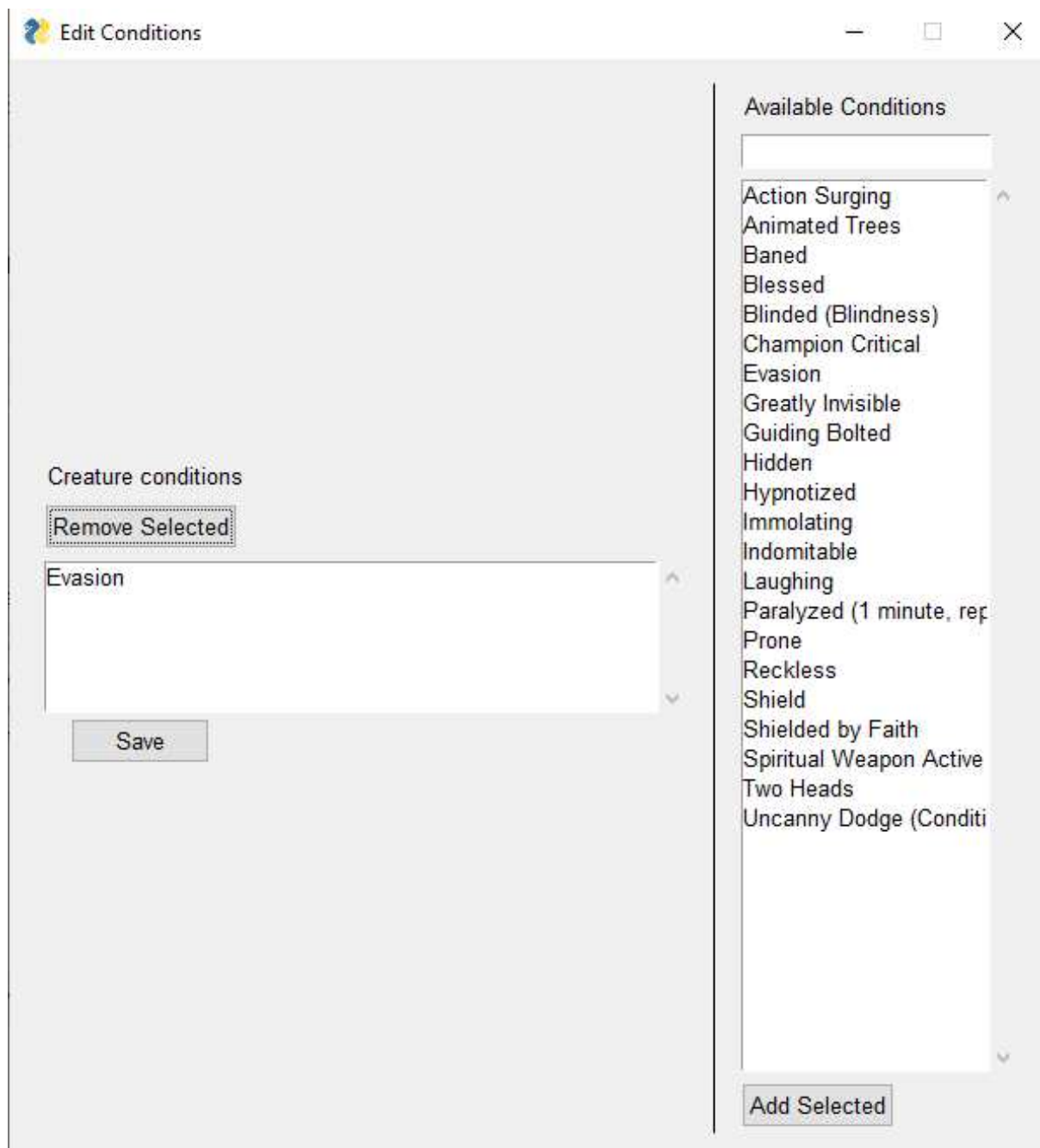


Figura 11 – Tela para adicionar condições permanentes aos atores.

#### 5.4.2.3 Conditions

Condições geralmente são causadas pelo efeito de ações, mas existem algumas habilidades, como evasão de monges e ladinos, que modificam aspectos dos atores de forma semelhante a uma condição. Essa tela permite que sejam adicionadas condições permanentes aos atores, que estão sempre ativas.

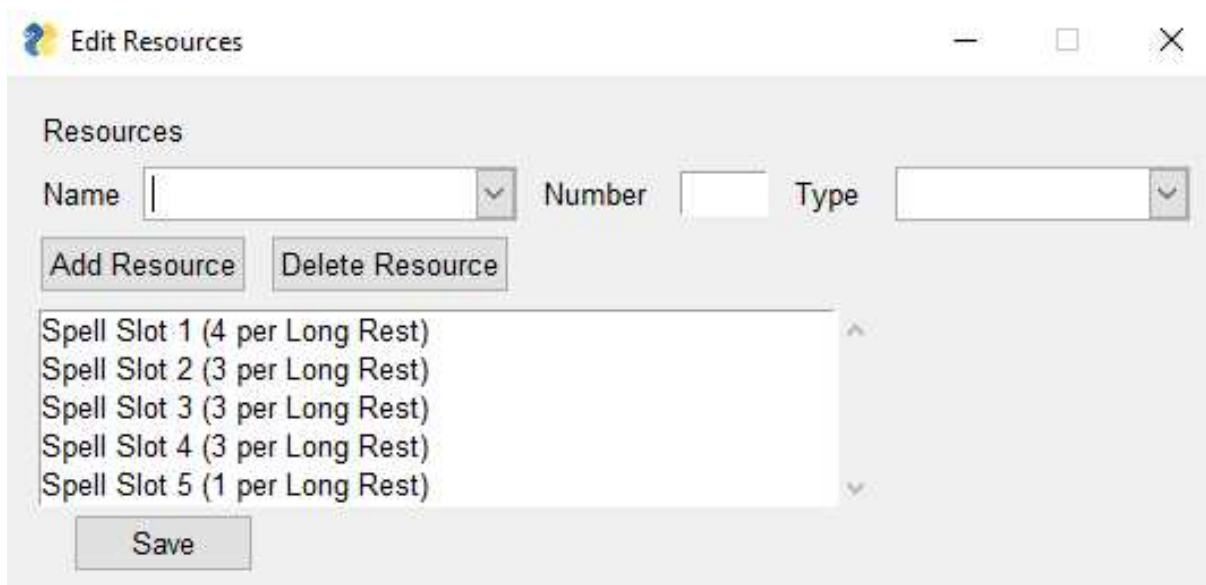


Figura 12 – Tela para edição de recursos disponíveis aos atores.

#### 5.4.2.4 Resources

Recursos são um aspecto fundamental do jogo, limitando a quantidade de vezes que os atores podem tomar certas ações. Para o simulador, cada ator possui sua própria lista de recursos, que vai diminuindo ao longo do tempo em que o ator utiliza os recursos. Cada recurso possui um nome, que é também sua identificação, uma quantidade máxima e um tipo.

Os tipos de recurso podem ser *Long Rest*, *Short Rest*, *Recharge X* ou *Start of Turn*, definindo o que é necessário para recuperar os recursos. *Long Rest* significa que é necessária uma noite de descanso para recuperar o recurso; *Short Rest* significa que é necessária uma hora de descanso para recuperar o recurso; *Recharge X*, em que X é um número entre 1 e 6, indica que após o recurso ser gasto, no começo do turno do ator, será rolado 1d6, e o recurso será recuperado se o valor rolado for maior ou igual ao valor de X; e *Start of Turn*, que significa que o recurso é recuperado no início de cada turno (isso é efetivamente idêntico a *Recharge 1*). O simulador recupera os recursos em combate como nas regras, e recupera os recursos de *Short Rest* opcionalmente quando realizando múltiplos combates em seguida.

#### 5.4.2.5 Combos

Existem atores que podem realizar diversas ações, completamente independentes, em um único turno. Os exemplos mais comuns disso são os personagens que lutam com armas que tem a habilidade *Extra Attack*, permitindo que ataquem mais de uma vez em um turno. Múltiplos monstros possuem a ação semelhante *Multiattack*, permitindo que realizem uma sequência definida de ações uma atrás da outra.

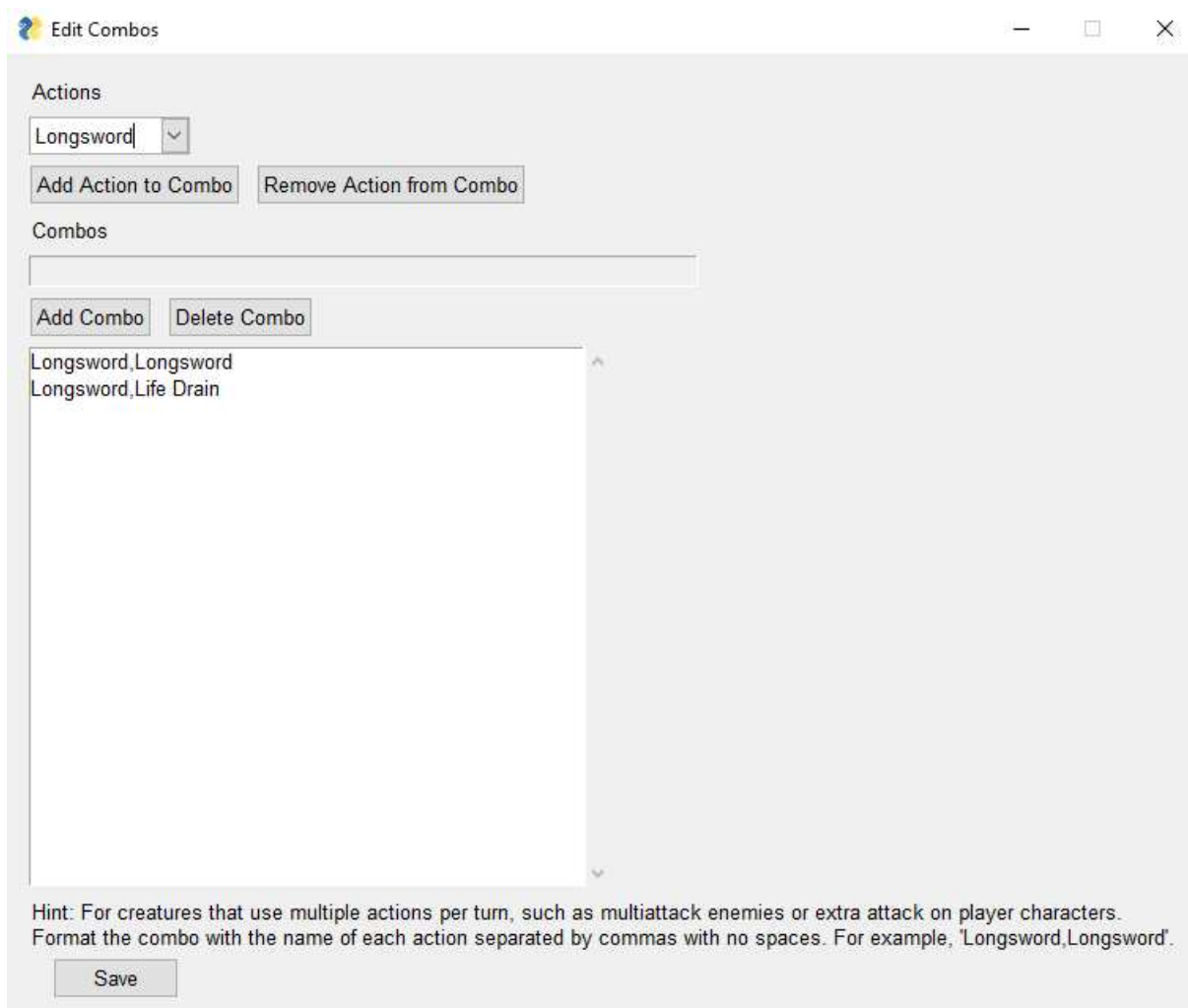


Figura 13 – Tela para edição de combos dos atores.

Nessa tela, é possível definir sequencias de ações a serem tomadas que permitem imitar essas habilidades. Para definir um combo, deve ser escrita o nome de cada ação separado por vírgulas, sem espaços. Por exemplo, um Bárbaro que ataca com um machado duas vezes teria o combo “Machado,Machado”.

### 5.4.3 Action

As ações são o ponto fundamental que permitem que atores interajam entre si. Nessa aba, é possível criar e editar as ações, separadas em três telas: *Action Stats*, *Costs* e *Effects*

#### 5.4.3.1 Action Stats

Essa tela permite definir os aspectos básicos necessários para cada ação, incluindo seu nome, velocidade, quantidade de alvos e tipo de alvos.

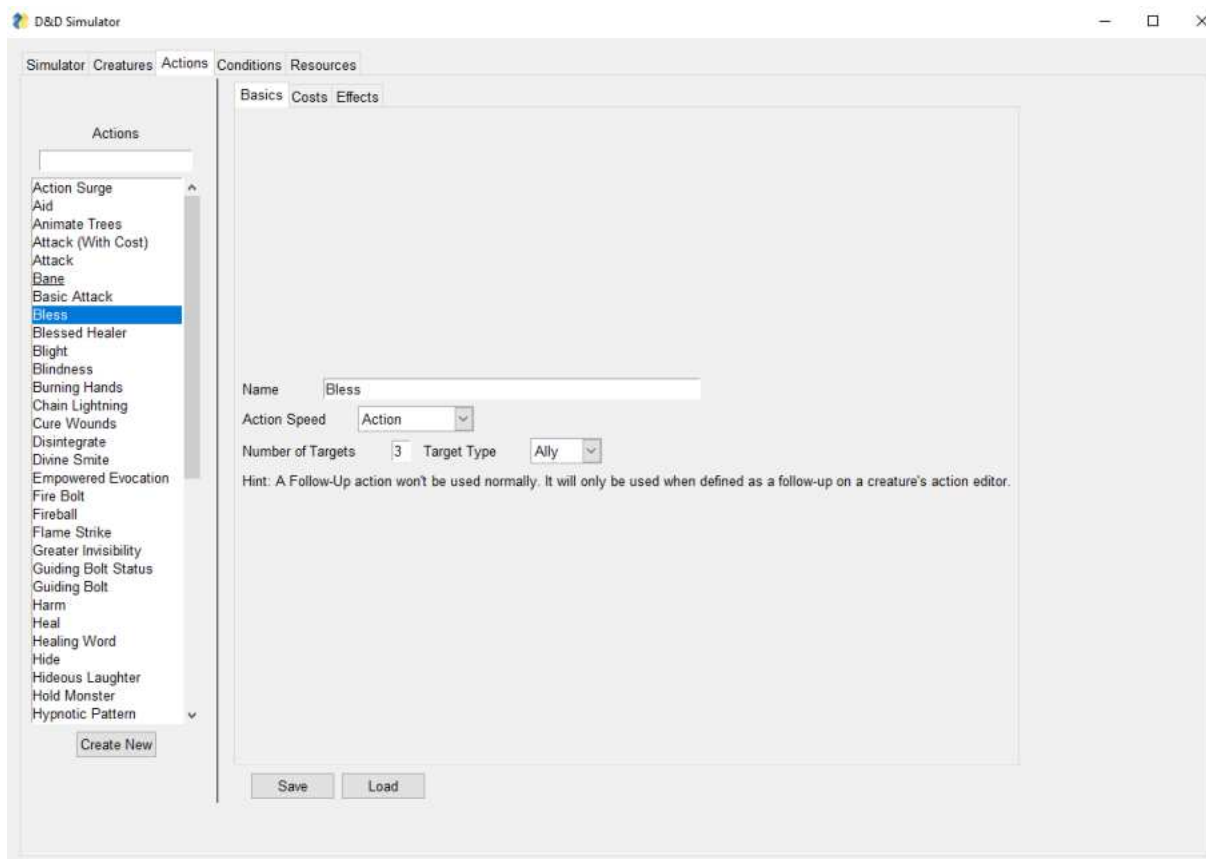


Figura 14 – Tela de edição dos atributos base da ação.

Velocidade pode ser *Action*, *Bonus Action*, *Free Action* ou *Follow-Up*. *Action* e *Bonus Action* são os tipos de ações comuns presentes no jogo. *Free Action* são ações como *Action Surge* e *Reckless Attack*, que podem ser feitas sem consumir sua ação ou ação bônus. *Follow-up* são ações que não podem ser usadas diretamente, e são chamadas como partes secundárias de outras ações. Exemplos incluem *Divine Smite* e *Stunning Strike*.

Em relação aos alvos, o tipo de alvo pode ser *Enemy*, *Ally* ou *Self*, indicando que pode afetar inimigos, aliados, ou somente o próprio usuário. Para o caso de inimigos ou aliados, pode ser definido um valor maior que 1 para a quantidade de alvos, indicando que é um efeito que atinge uma área.

#### 5.4.3.2 Costs

Essa tela corresponde à seção de recursos das criaturas. Naquela seção foram definidos quais recursos a criatura tem acesso, e nessa seção é definido o como cada ação consome esses recursos, de maneira semelhante à seção das criaturas.



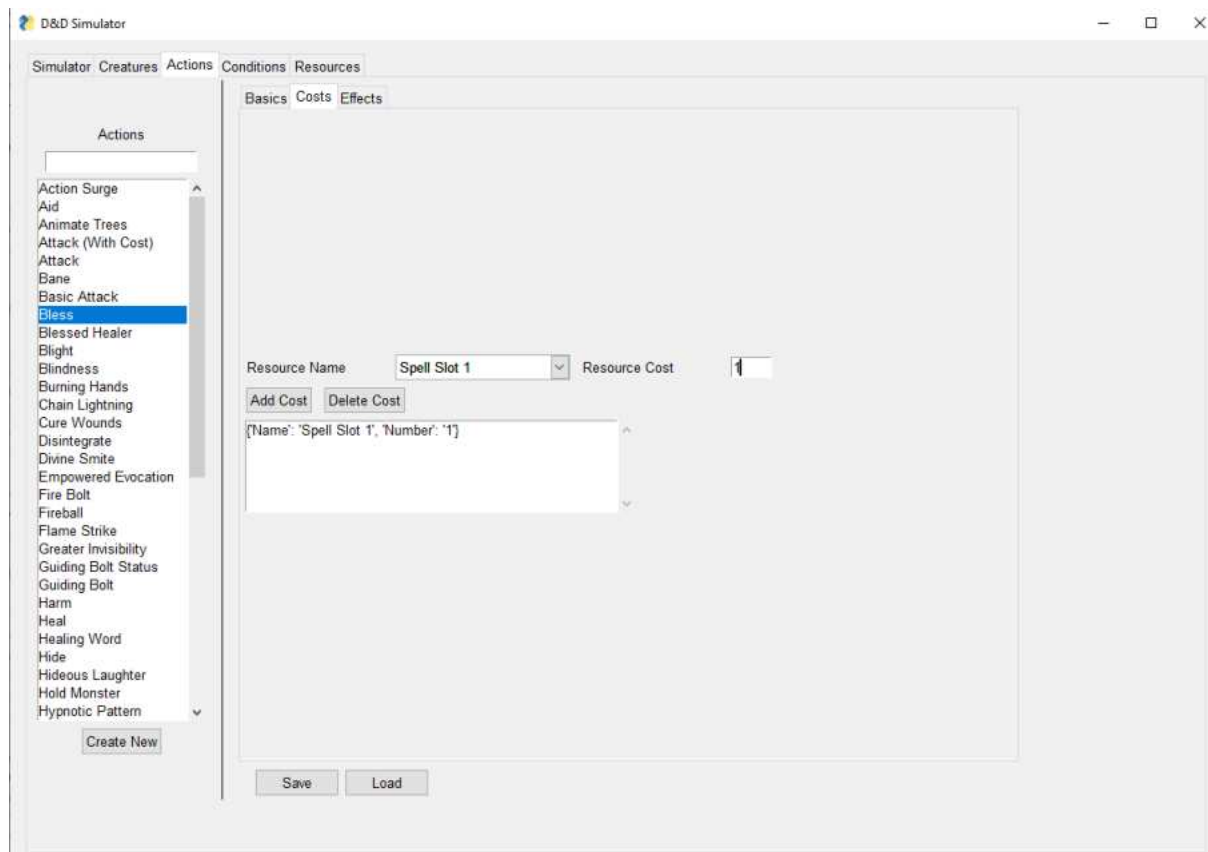


Figura 15 – Tela de edição dos custos das ações.

#### 5.4.3.3 Effects

Aqui pode ser definido o que a ação deve fazer. Tem dois aspectos importantes a serem definidos aqui: O tipo de tentativa, e o tipo de efeito, como descrito anteriormente. Após definidos, menus adicionais aparecem, detalhando diferentes aspectos correspondentes às escolhas. A tentativa *Attack Roll* permite que seja definido um valor para o bônus de ataque, enquanto o efeito *Damage* permite que seja definido uma ou mais rolagens de ataques, com tipos de dano especificados.

#### 5.4.4 Condition

Nessa aba são criadas e editadas as condições, todo tipo de efeito que altera ou afeta um ator durante o combate. Essa aba é dividida nas telas *Condition Stats*, *Offense Modifiers*, *Defense Modifiers*, *Economy Modifiers* e *Over Time*

##### 5.4.4.1 Condition Stats

Essa tela permite a definição dos aspectos comuns a toda condição. Esses incluem seu nome e duração (em turnos, tipicamente 1, 2, 10 ou 100 turnos), além de dois mais detalhados, a forma de término e se é uma condição pareada.

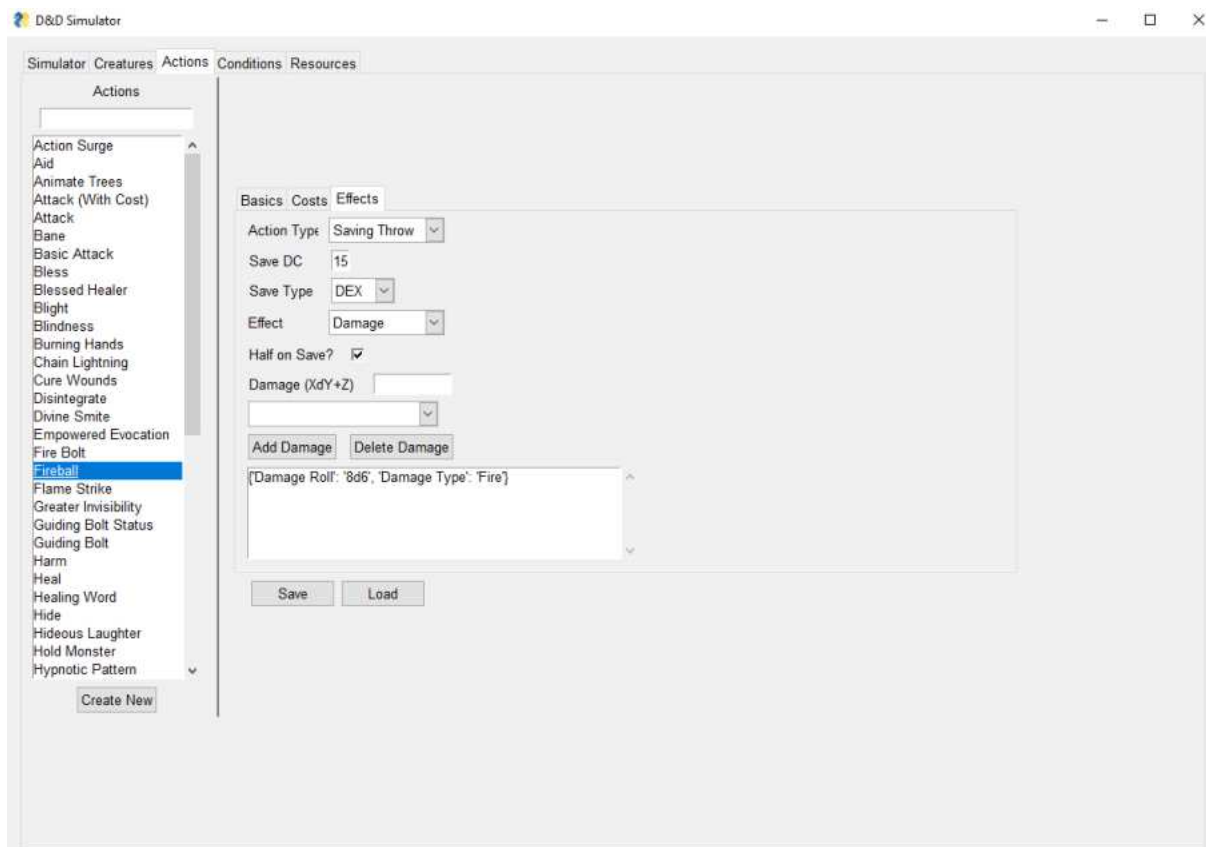


Figura 16 – Tela para edição dos efeitos das ações.

A forma de término indica em qual momento que a duração da condição progride, até chegar em zero e acabar. Isso pode ser no começo ou no fim do turno do ator que causou a condição ou do ator sendo afetado pela condição. Além disso, existem formas que causam o término mais rápido da condição. Isso inclui condições que acabam quando o alvo recebe dano, ou que fazem o alvo repetir uma salvaguarda que foi feita para resistir ao efeito inicialmente no fim do seu turno.

Condições pareadas são uma nomenclatura que foi dada para condições que afetam uma criatura somente em relação à criatura que causou o efeito, ou vice-versa. Por exemplo, *Hunter's Mark* é uma magia que afeta um alvo, causando uma condição que faz esse alvo receber dano extra dos ataques do ator que usou a magia. Para esses casos, a condição é considerada como ativa somente quando esses atores interagem entre si, e os efeitos dela são aplicados no ator que causou a condição.

#### 5.4.4.2 Offense Modifiers

Aqui são especificados todos os modificadores relacionados a ataques do ator sendo afetado. Pode incluir um aumento ou redução do bônus de ataque, forçar ataques a serem feitos com vantagem ou desvantagem, modificar o primeiro dano de um ataque por um valor inteiro especificado, adicionar uma rolagem de dano inteira a mais, mesmo de um

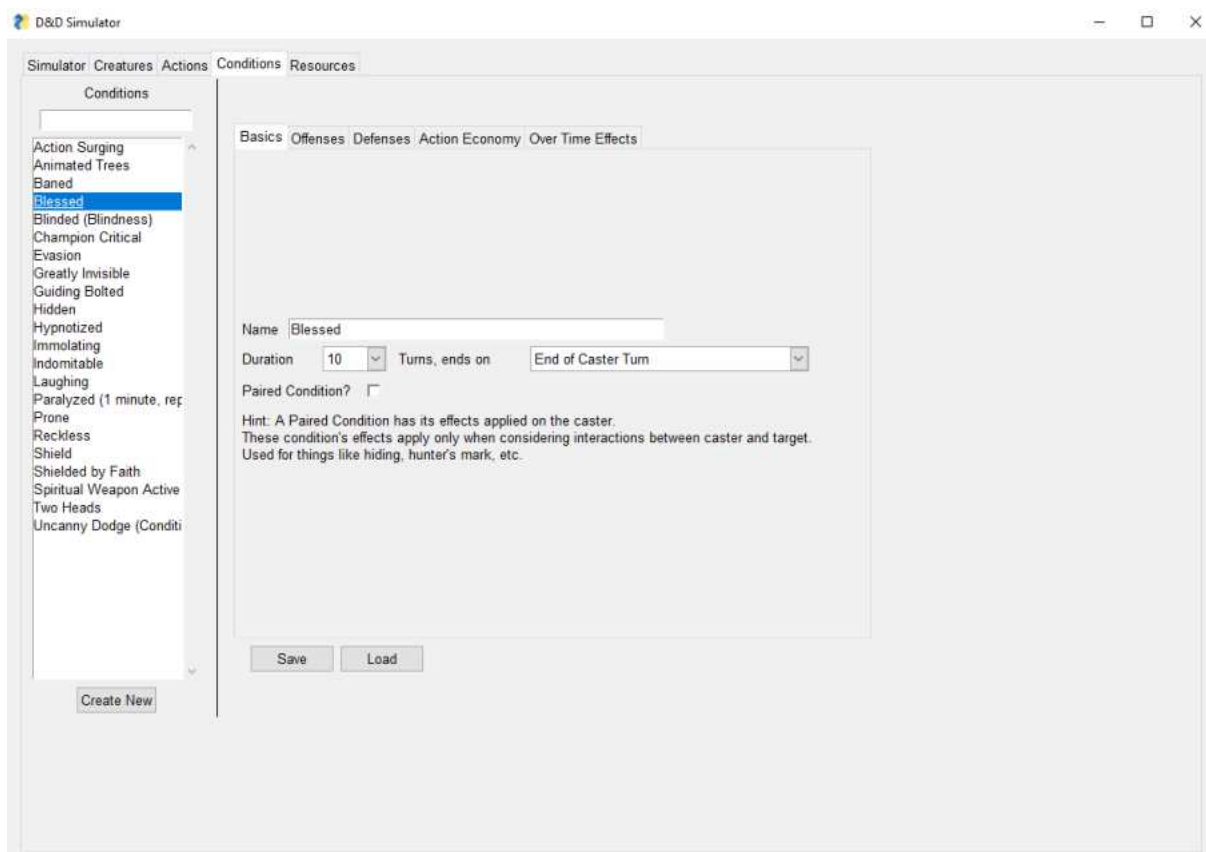


Figura 17 – Tela para edição dos atributos básicos de condições

tipo diferente, e modificar o valor necessário para um acerto crítico, que normalmente é 20.

#### 5.4.4.3 Defense Modifiers

Aqui, por sua vez, são editados os modificadores da defesa do ator afetado. Podem incluir um aumento ou redução da classe de armadura, forçar ataques contra o afetado a serem feitos com vantagem ou desvantagem, fazer com que qualquer ataque que acerte o alvo seja um crítico, ou inclusive adicionar resistências, vulnerabilidades ou imunidades.

Também é possível adicionar modificações para cada um dos seis tipos de salvaguardas, individualmente. Isso pode incluir modificadores numéricos ou de vantagem/desvantagem, como na classe de armadura, mas também pode ser dada uma mecânica chamada evasão para a salvaguarda. Em grande parte dos efeitos que requerem uma salvaguarda, falhar resulta em sofrer o dano completo do efeito, enquanto passar na salvaguarda reduz o dano tomado pela metade. No entanto, quando um ator possui evasão, ele recebe metade do dano quando falha, e ignora completamente quando passa.

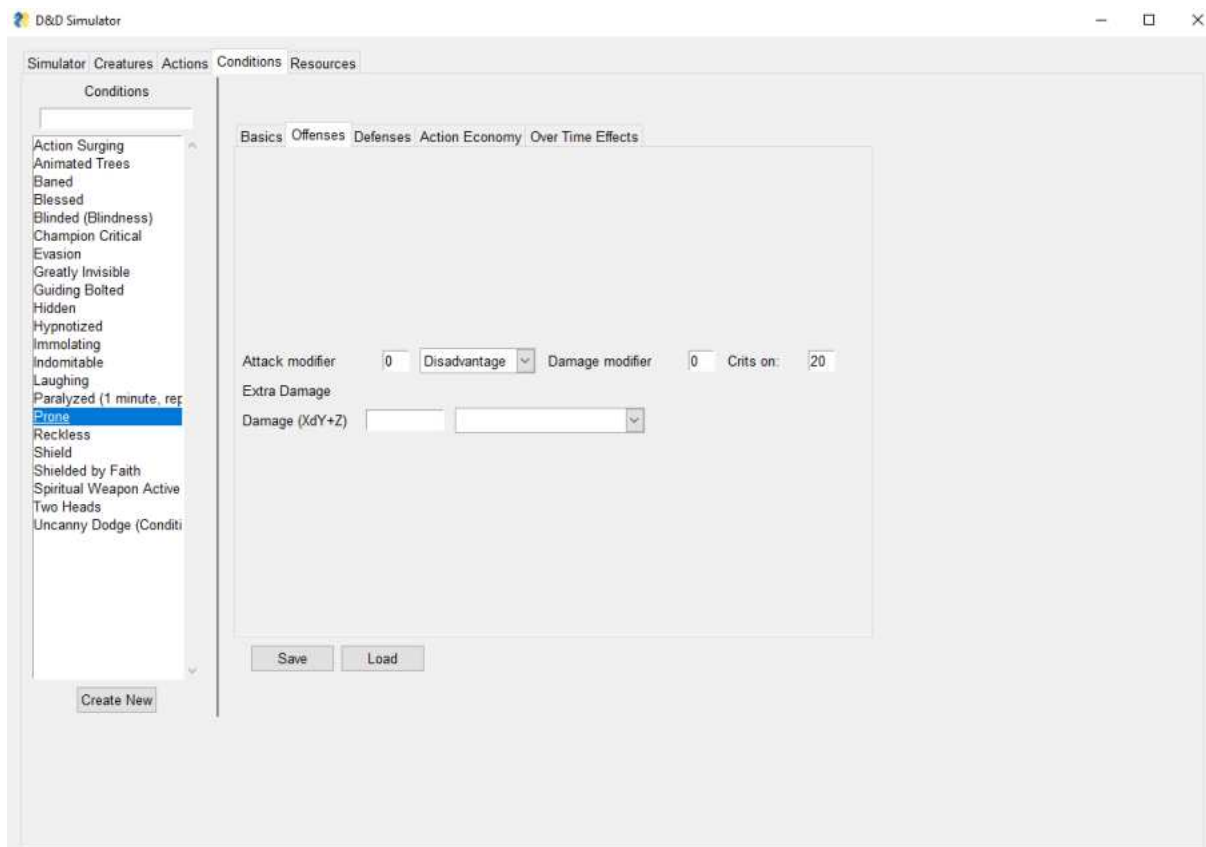


Figura 18 – Tela para edição dos modificadores de ataque da condição

#### 5.4.4.4 Economy Modifiers

Essa tela foi nomeada em relação a um conceito de jogos de RPG conhecido como a economia de ações, que diz que em um combate, o lado com mais ações tende a vencer. Condições com modificadores de economia podem aumentar ou reduzir a quantidade de ações, de qualquer tipo, do ator. Exemplos incluem *Action Surge*, que permite que o ator tenha uma ação a mais, ou a condição de incapacitado, que impede completamente o ator de tomar ações.

#### 5.4.4.5 Over Time

Alguns efeitos, embora raros em D&D, incluem recuperação de vida ou dano que ocorre todo turno do ator afetado, repetidamente. Aqui podem ser implementados efeitos desse tipo de forma simples.

#### 5.4.5 Resources

A aba de *Resources* é somente uma tela simples para definição de nomes de recursos, permitindo adicionar novos ou remover aqueles que não estão em uso. Os recursos salvos

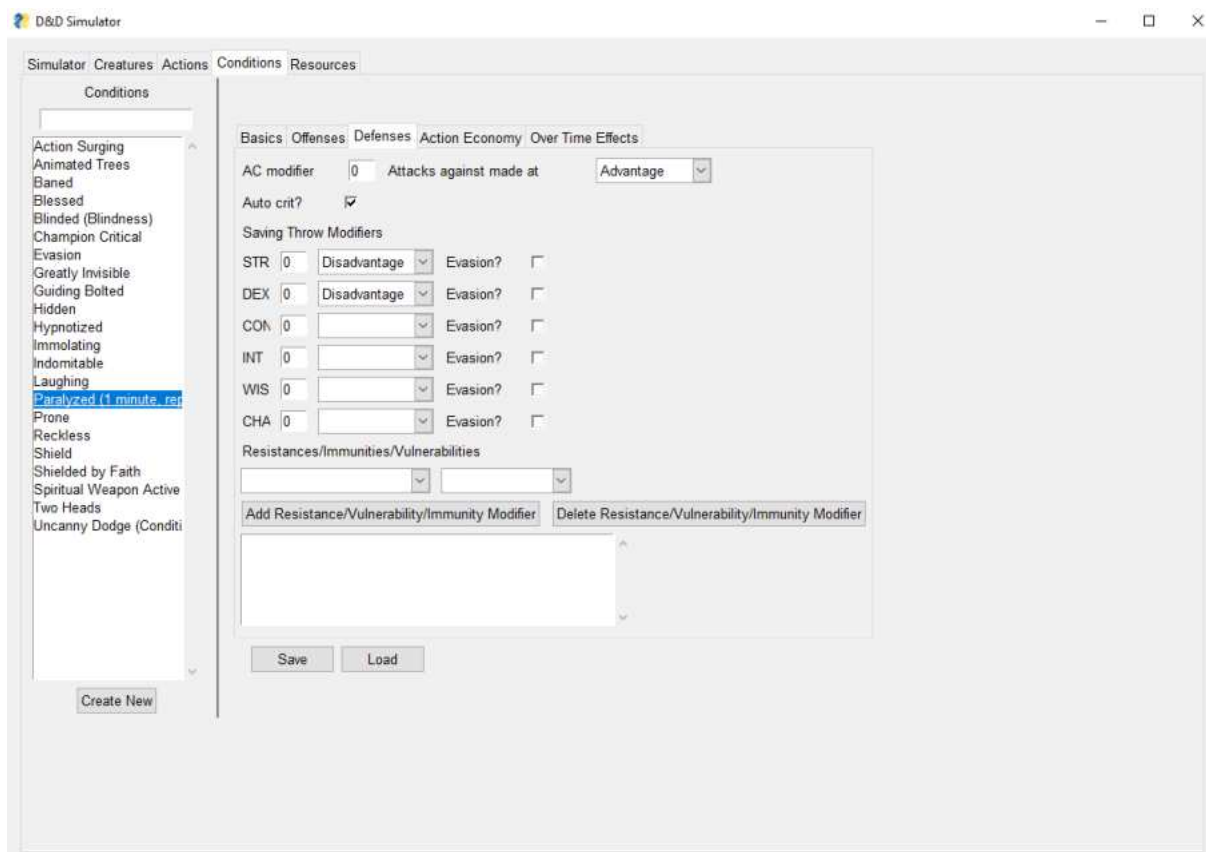


Figura 19 – Tela para edição dos modificadores de defesa da condição

aqui são os mesmos que estarão disponíveis nas telas de edição de recursos de criaturas e de ações.

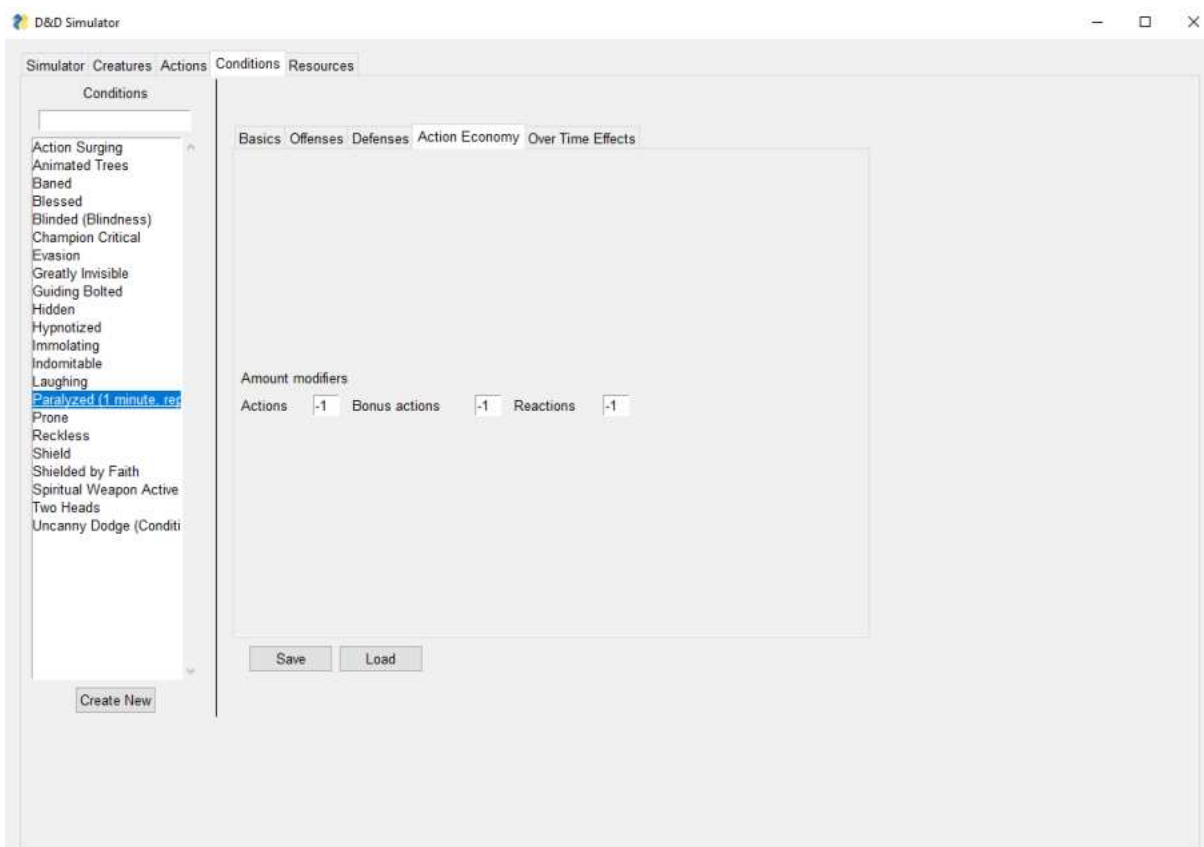


Figura 20 – Tela para edição dos modificadores da economia de ações da condição.

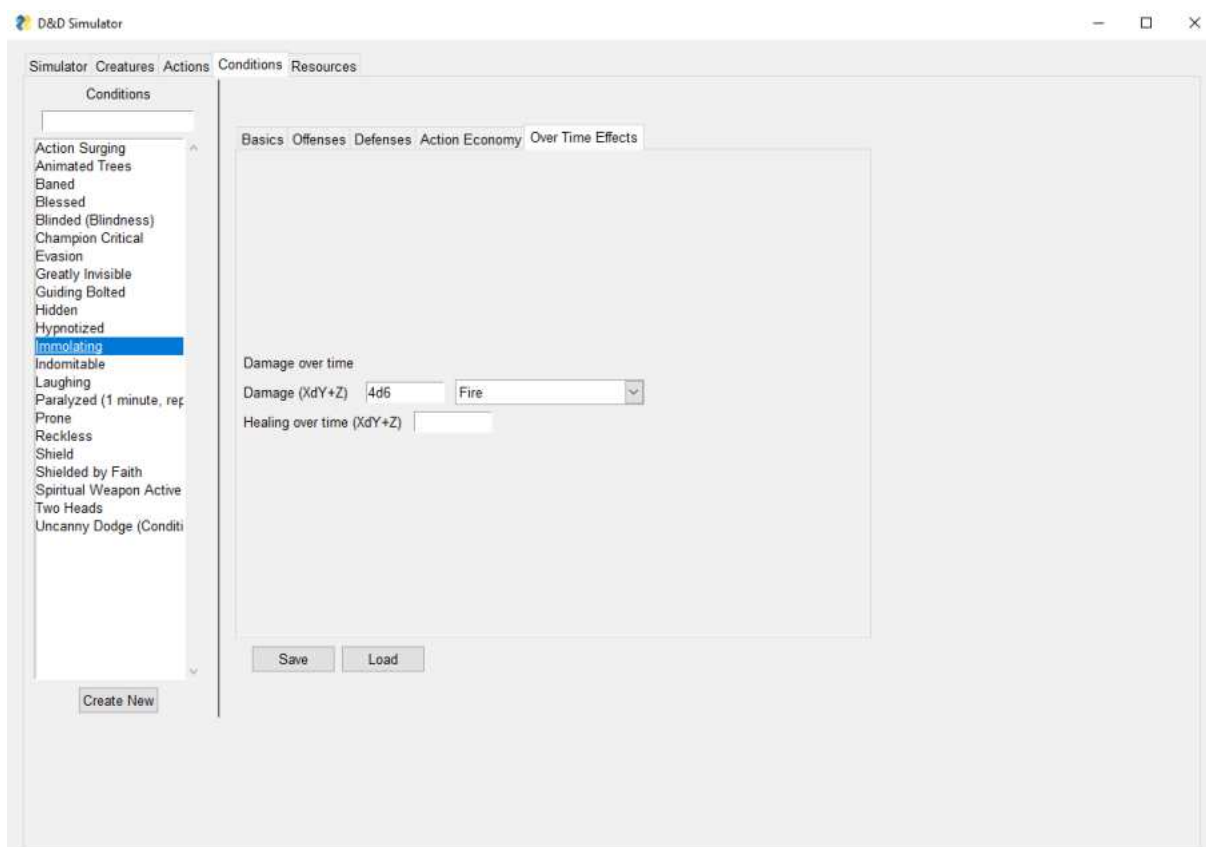


Figura 21 – Tela para edição de efeitos da condição que ocorrem ao longo do tempo.

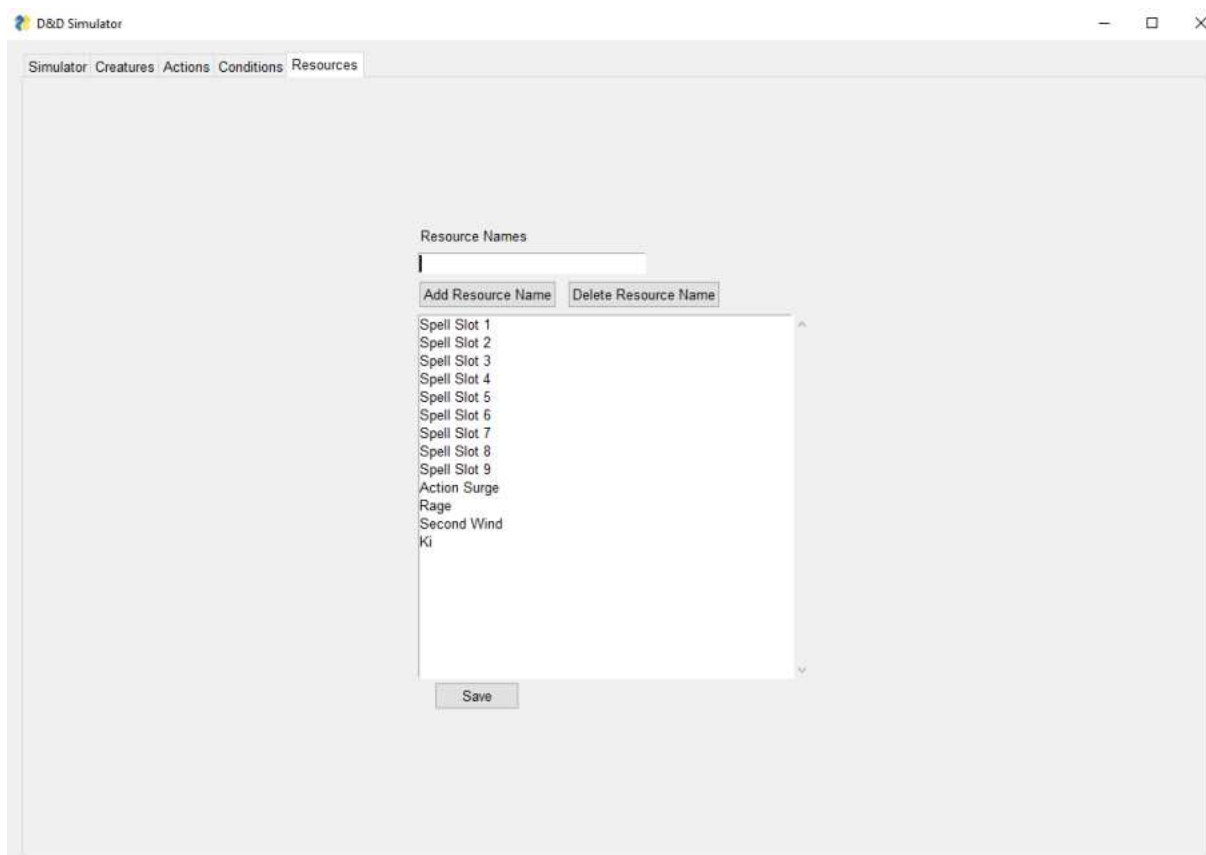


Figura 22 – Tela para definição de nomes de recursos.



## 6 AVALIAÇÃO

### 6.1 AVALIAÇÃO DA API

#### 6.1.1 Testes de Desempenho

Para os testes de desempenho, foi calculado o tempo que leva desde o usuário apertar o botão de "simular" na aba de simulação, até os resultados da simulação serem devolvidos para o usuário, considerando um total de 1000 simulações. Diversos fatores influenciam nessa duração, especialmente a quantidade e complexidade de atores sendo simulados. Além disso, é importante destacar também que é possível realizar as simulações com ou sem o passo-a-passo registrado para visualização de combates individuais. Nesse caso, o tempo de espera é significativamente elevado.

As figuras 23 e 24 mostram o tempo de duração de combates obtidos em simulações de teste. Foram simulados 1000 combates entre um ator contra um ator, dois atores contra um ator, dois atores contra dois atores, e assim por diante, como mostrado nos gráficos. Foi realizado um teste com times compostos por Orcs, uma criatura bem conhecida e simples, sem ações muito complexas, e *Wizards* de nível 9, que possuem uma complexidade maior em termos de suas ações e mais pontos de vida.

Foi observado que aumentar a quantidade de criaturas em um time que está na vantagem para ganhar não causa um aumento significativo na duração da execução. Os maiores aumentos ocorrem quando um time em desvantagem (ou ambos times, em caso de igualdade) recebem atores adicionais, pois o time em vantagem terá que realizar mais ataques

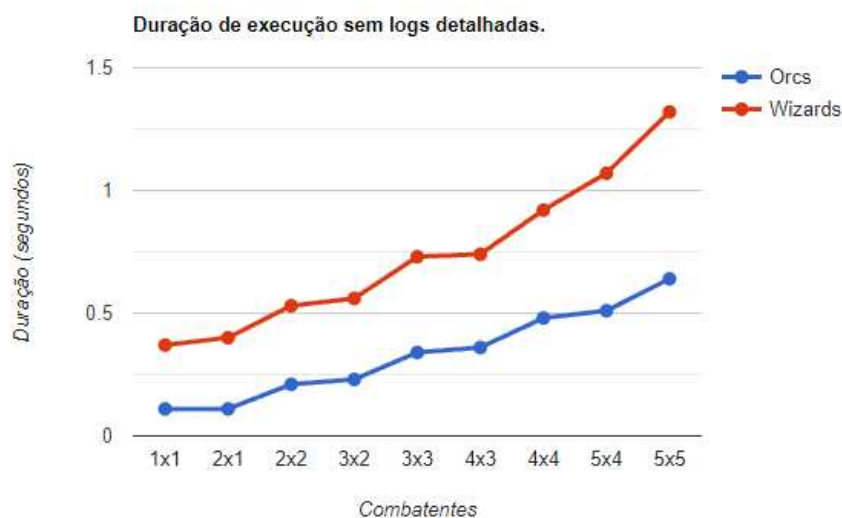


Figura 23 – Duração de simulação sem logs detalhadas

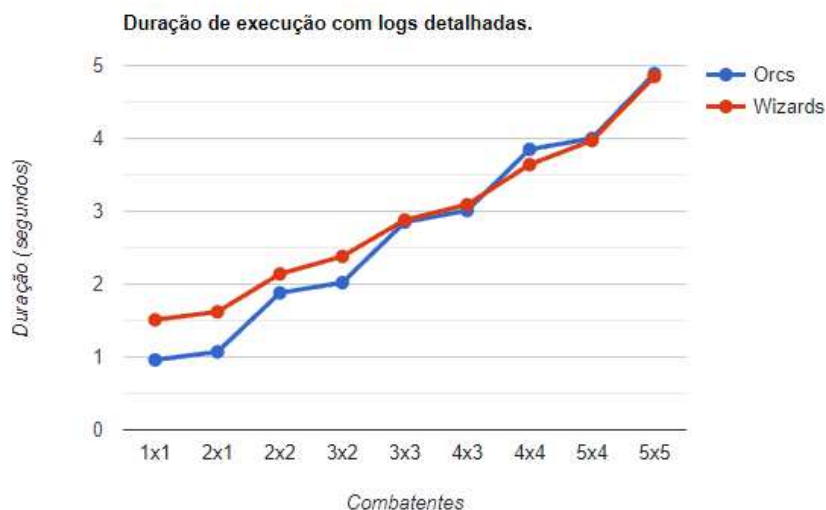


Figura 24 – Duração de simulação com logs detalhadas

para derrotar todos os inimigos. Em uma escala de quantidade de inimigos próxima ao que se espera encontrar em uma campanha real de D&D, observou-se um aumento quase linear da duração.

### 6.1.2 Testes de Sanidade

Para testes de sanidade, foram feitos alguns testes mais específicos para garantir a funcionalidade dos sistemas implementados.

#### 6.1.2.1 Partida Espelho

Uma partida espelho é um termo utilizado para quando ambos lados em um combate são exatamente iguais. Nesse caso, ambos lados tem chance igual de vencer, e portanto, a porcentagem de vitórias de cada lado em um combate simulado deve ser extremamente próximo a 50%. Foram realizadas 10000 simulações de partidas espelho, utilizando Orcs, *Wizards* de nível 9, Treants, Ogres e Owlbears, todos criaturas ou personagens criáveis por sistemas oficiais de D&D. A figura 25 mostra os resultados dessa simulação, demonstrando que todos os resultados foram extremamente próximos de 50%, que é o resultado esperado.

#### 6.1.2.2 Combate altamente desbalanceado

Mesmo com boa sorte, uma desvantagem de atributos extremamente elevada é incrivelmente difícil de ser ultrapassada. O propósito desse teste é mostrar que, em um combate entre uma criatura fraca e outra extremamente forte, a criatura forte deve sempre vencer. Para esse teste, foram selecionados um *Fighter*, *Wizard*, *Cleric* e *Rogue*, todos de nível 1, e

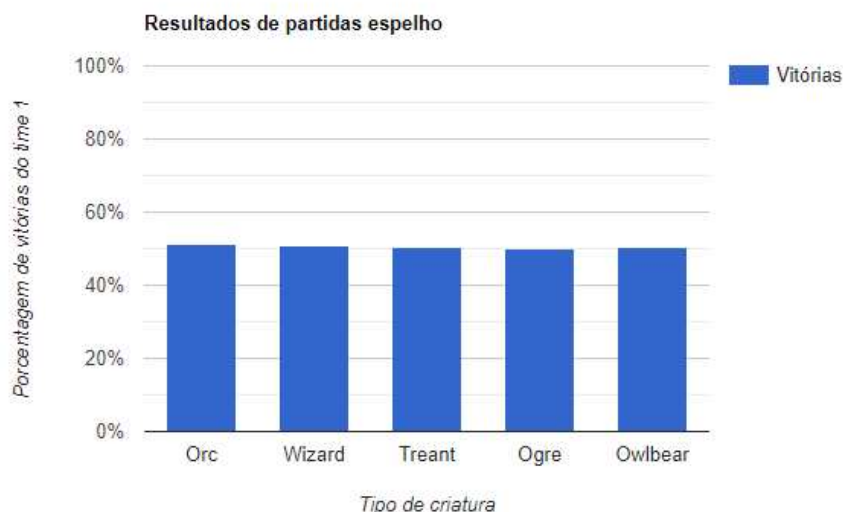


Figura 25 – Partida espelho entre diversos tipos de criaturas diferentes.

forçados a lutar contra um personagem da mesma classe de nível 11 em 10000 simulações de combate. Como esperado, os personagens de nível 11 venceram 100% dos combates.

#### 6.1.2.3 Combate com ator imune

Para testar as mecânicas relacionadas a alteração de dano recebido, foi realizada 1000 simulações de combate entre um Orc e um Stone Giant, em que o Stone Giant foi o vencedor 100% das vezes. Em seguida, o Orc foi alterado para possuir imunidade a dano físico, o único tipo de dano que o Stone Giant é capaz de causar. Após essa alteração, o Orc venceu 100% dos combates, como esperado.

### 6.1.3 Comparação com o Modelo Oficial do Dungeons & Dragons

No livro de referencia para os DMs, *Dungeon Master's Guide*, existem regras definidas sobre o como criar encontros de combate com dificuldades especificadas como fácil, médio, difícil e fatal. A dificuldade é calculada somando a quantidade de pontos de experiência de cada monstro, multiplicando por um valor dependente da quantidade de oponentes, e comparando ao nível dos personagens. Utilizando o simulador, foram construídos diversos encontros de combate para análise.

#### 6.1.3.1 Utilizando encontros definidos manualmente

Primeiramente, foram definidos manualmente cinco níveis de personagens para serem analisados, sendo os níveis 1, 3, 5, 9 e 11. Esses níveis especificamente foram escolhidos pois representam aumentos significativos do poder dos personagens, com o nível 1 sendo o nível inicial, o nível 3 sendo o nível em que ganham acesso a suas subclasses, o nível

5 e 9 sendo níveis em que recebem aumento de proficiência, e o nível 11 sendo o início do “Tier 3” do jogo. Níveis acima do nível 11 não foram analisados detalhadamente, pois sofrem um aumento elevado de complexidade, e há um volume baixo de campanhas que chegam até esses níveis.

Para cada um desses cinco níveis, foram criados quatro encontros correspondentes aos quatro níveis de dificuldade, seguindo as regras oficiais, utilizando apenas monstros disponíveis publicamente no *Systems Reference Document*. Os encontros foram:

- Nível 1
  - Fácil: 1x Orc
  - Médio: 4x Merfolk
  - Difícil: 2x Orc
  - Fatal: 4x Skeleton
  
- Nível 3
  - Fácil: 2x Orc
  - Médio: 2x Ghoul
  - Difícil: 1x Ghoul + 5x Zombie
  - Fatal: 4x Animated Armor
  
- Nível 5
  - Fácil: 5x Orc
  - Médio: 5x Ghoul
  - Difícil: 3x Ogre + 3x Goblin
  - Fatal: 3x Wight + 2x Zombie
  
- Nível 9
  - Fácil: 2x Owlbear
  - Médio: 3x Wight + 2x Zombie
  - Difícil: 3x Minotaur + 2x Ogre
  - Fatal: 2x Stone Giant
  
- Nível 11
  - Fácil: 2x Ogre + 2x Bugbear + 2x Orc
  - Médio: 3x Minotaur + 2x Ogre

Tabela 1 – Resultados da avaliação com encontros criados manualmente

Nível	Vitória (%)				PV restante (%)			
	Fácil	Médio	Difícil	Fatal	Fácil	Médio	Difícil	Fatal
1	100%	99.8%	99.5%	94%	95.9%	80.1%	86.9%	65.1%
3	100%	100%	100%	93%	97.2%	93.5%	84.7%	74.7%
5	100%	100%	100%	100%	97.1%	92.7%	87.5%	90.9%
9	100%	100%	99.4%	100%	96%	94.5%	76%	81.5%
11	100%	100%	100%	99.9%	94.7%	84.1%	89.1%	86.1%

- Difícil: 2x Stone Giant
- Fatal: 1x Treant + 2x Ettin

Para representar o lado dos jogadores, foram criados quatro personagens, cujas raças e classes foram definidas por escolhas comuns por partes dos jogadores e limitados pelo que é disponível no *Systems Reference Document*. O grupo de 4 foi composto por um *Human Fighter (Champion)*, um *High Elf Wizard (Evocation)*, um *Hill Dwarf Cleric (Life)*, e um *Lightfoot Halfling Rogue (Thief)*.

A tabela 1 mostra os resultados de 1000 simulações de combate para cada uma das lutas, com a porcentagem de vitórias do time dos jogadores, e a quantidade média de pontos de vida restantes que tiveram quando venceram. É possível ver pelos resultados que realmente houve um aumento na dificuldade dos encontros, especialmente quando se vem aos pontos de vida restantes. Curiosamente, encontros com quantidades maiores de inimigos fracos tendem a causar perda de vida maior que encontros com quantidades menores de inimigos mais fortes, mesmo após ajustando os valores de experiência de acordo com as regras oficiais.

Observa-se que, em nenhuma das lutas, o lado dos jogadores teve chance abaixo de 90% de vencer. Isso faz sentido, pois D&D é um jogo com grande foco em sua história e personagens, e uma perda total frequentemente significa o fim da história. Por outro lado, os níveis mais baixos, como nível 1 e 3, mostraram uma frequência surpreendentemente grande de perdas em encontros fatais, comprovando uma fala comum que diz que os níveis iniciais são os mais perigosos.

Os testes mostram uma clara relação positiva entre os cálculos presentes no *Dungeon Master's Guide* e as simulações realizadas. Assim, é possível afirmar que a ferramenta tem capacidade de auxiliar o DM em sua preparação de encontros de combate, permitindo que teste a dificuldade com simulações de situações reais e obtenha múltiplas métricas adicionais que podem ser úteis para o que deseja, incluindo a duração média do combate, quais personagens tem chances maiores de serem derrotados e o quanto o encontro afetará os personagens para os próximos encontros.

### 6.1.3.2 Utilizando encontros definidos de forma aleatória

Além dos encontros definidos manualmente, foram também criados encontros definidos de forma aleatória. Para tanto, foram utilizadas ferramentas disponíveis online para gerar um grupo de personagens aleatórios, e diversos encontros aleatórios para esses personagens. Esse tipo de ferramenta vê bastante utilização hoje em dia, então é uma métrica interessante para ser observada.

Os personagens criados foram um *Red Dragonborn Paladin (Devotion)*, um *Half-Orc Barbarian (Berserker)*, um *Half-Orc Monk (Open Hand)* e um *Tiefling Sorcerer (Red Draconic Bloodline)*. Já os encontros definidos foram os seguintes:

- Nível 1
  - Fácil: 1x Crocodile
  - Médio: 1x Imp
  - Difícil: 1x Orc + 1x Hobgoblin
  - Fatal: 1x Plesiosaurus
- Nível 3
  - Fácil: 2x Giant Bat + 2x Giant Crab
  - Médio: 2x Bugbear
  - Difícil: 1x Tiger + 4x Giant Wolf Spider
  - Fatal: 2x Giant Spider + x2 Hippogriff
- Nível 5
  - Fácil: 1x Ettin
  - Médio: 1x Elephant + 1x Tiger
  - Difícil: 1x Giant Ape
  - Fatal: 1x Wyvern + 1x Minotaur
- Nível 9
  - Fácil: 1x Chimera
  - Médio: 2x Veteran + 4x Bugbear
  - Difícil: 2x Ankylosaur + 4x Rhinoceros
  - Fatal: 1x Stone Giant + 4x Berserker
- Nível 11

Tabela 2 – Resultados da avaliação com encontros gerados aleatoriamente

Nível	Vitória (%)				PV restante (%)			
	Fácil	Médio	Difícil	Fatal	Fácil	Médio	Difícil	Fatal
1	100%	99.9%	99.9%	86.1%	95.0%	90.4%	84.5%	68%
3	100%	100%	99.4%	99.4%	88.2%	87.6%	75.4%	79%
5	100%	100%	99.9%	99.7%	95.6%	88.5%	86%	81.9%
9	100%	100%	100%	100%	91.7%	86.9%	78.6%	80.2%
11	100%	100%	99.9%	99%	93%	88.9%	72%	58.3%

- Fácil: 2x Elephant
- Médio: 1x Giant Crocodile + 2x Manticore
- Difícil: 1x Storm Giant
- Fatal: 1x Young Bronze Dragon + 2x Fire Elemental

O experimento então foi realizado com esses combates seguindo as mesmas regras definidas na seção anterior.

A tabela 2 apresenta os resultados encontrados para os combates após a simulação. Observa-se que foram próximos aos obtidos durante o experimento com encontros definidos manualmente, indicando de forma geral um aumento na dificuldade do encontro simulado quando a dificuldade do encontro de acordo com as regras aumenta.

Existem inúmeras possíveis combinações de grupos de personagens e inúmeras possíveis combinações de inimigos para esses grupos enfrentarem. Com essa ferramenta, é possível que DMs possam planejar e compreender melhor os encontros de combate que desejam apresentar para seus jogadores.

## 6.2 AVALIAÇÃO QUALITATIVA COM JOGADORES

Para a avaliação qualitativa, foi considerada a norma ISO/IEC 25010 (ISO/IEC 25010, 2011), com maior importância dada para as categorias “adequação funcional”, “eficiência de desempenho” e “usabilidade”, que são mais relevante ao projeto. Com base nessa norma, foram criadas afirmações (para uso com Escala de Likert) e perguntas que foram apresentadas a um total de 4 DMs experientes que utilizaram e experimentaram com a ferramenta.

Inicialmente, foram realizadas entrevistas individuais, em que foram realizadas perguntas abertas com o objetivo de receber *feedback* direto dos usuários para realizar melhorias na ferramenta. Múltiplas entrevistas foram realizadas alternativamente com alterações na ferramenta. As perguntas realizadas durante essas entrevistas são enumeradas a seguir.

- A ferramenta é útil?
- Você consegue usar sem eu te ensinar?

- A interface é boa?
- Como você vê a utilidade da ferramenta?
- O que você gostou?
- O que você não gostou?

A partir das respostas obtidas durante essas entrevistas, diversas melhorias foram realizadas, especialmente nos aspectos da usabilidade da interface. Após esse período, os usuários que experienciaram a versão final da ferramenta responderam a afirmações em modelo de Escala de Likert, em escala de 1-5. As afirmações, a subcategoria da norma ISO/IEC correspondente e o valor médio obtido das respostas são listados a seguir.

- A ferramenta é útil para um DM planejar encontros. (Propriedade Funcional)
  - 5
- Os resultados da ferramenta estão corretos segundo sua experiência de DM. (Correção Funcional)
  - 4
- A interface é de fácil utilização. (Operabilidade)
  - 4.25
- Foi fácil aprender a utilizar as funcionalidades da ferramenta. (Capacidade de aprendizado)
  - 4
- A ferramenta possuiu todas as funcionalidades necessárias para simulação. (Integridade Funcional)
  - 3
- O tempo de resposta é adequado. (Comportamento do Tempo)
  - 4.75
- A ferramenta facilitou o planejamento de minhas sessões.
  - 4.25
- Eu recomendaria essa ferramenta para outros DMs.
  - 4.75

Observou-se um resultado majoritariamente positivo, com os usuários declarando que a ferramenta é útil para os propósitos de um DM, que é o principal objetivo da ferramenta.



## 7 CONCLUSÃO

Os resultados observados pelos testes quantitativos mostram que o simulador foi capaz de alcançar seus objetivos, servindo como uma ferramenta para DMs conseguirem analisar e ajustar encontros de combate, usando simulações capazes de seguir as regras oficiais de D&D e os dados obtidos através dessas simulações.

As avaliações quantitativas obtiveram resultados em linha com os resultados esperados pelas regras oficiais apresentadas no *Dungeon Master's Guide*, mostrando que as simulações se mantiveram consistentes com o que se espera em um cenário real. As avaliações qualitativas por sua vez mostraram que embora a ferramenta não esteja perfeita, ela se prova útil para a criação e planejamento de encontros no ponto de vista dos DMs.

Existem múltiplas possibilidades para avanços futuros na ferramenta, incluindo principalmente a implementação de alguns aspectos mais detalhados do jogo, como movimentação, mapas, alcances de ações e áreas de efeito “reais”. Houve alguns aspectos do combate que também não foram diretamente implementados, como reações que podem ser realizadas quando algum evento específico acontece; ações lendárias, específicas para alguns monstros capazes de usar ações fora de seu próprio turno; e habilidades, que tem sua utilidade principal fora de combate, mas podem ter seus usos durante. Finalmente, existem possibilidades adicionais para avaliações, incluindo por exemplo a comparação de resultados encontrados pela simulação com o resultado de encontros de combate realizados por jogadores humanos

Em conclusão, a ferramenta criada permite que DMs consigam preparar e equilibrar seus encontros de combate de forma mais detalhada e com mais dados, levando a uma experiência melhor para o DM e seus jogadores alheios.

## REFERÊNCIAS

- EDUNN, T. **The Finished Book**. Accessed: 2024-07-23. Disponível em: <https://tomedunn.github.io/the-finished-book/>.
- FERLA, M. **D&D Tools**. Accessed: 2024-07-23. Disponível em: <https://dnd.matteoferla.com/>.
- HAUGE, M.; STARR, M. Life's a party: Analyzing party compositions for combat encounters in dungeons & dragons fifth edition through monte carlo methods. 2021.
- HITCHENS, M.; DRACHEN, A. The many faces of role-playing games. **International Journal of Role-Playing**, n. 1, p. 3–21, Dec. 2008. Disponível em: <https://journals.uu.se/IJRP/article/view/185>.
- ISO/IEC 25010. **ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models**. 2011.
- JUNIOR, A. S.; SARINHO, V. Tarcos: Providing a combat simulation tool for tabletop role-playing games. In: **Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital**. Porto Alegre, RS, Brasil: SBC, 2021. p. 344–348. ISSN 0000-0000. Disponível em: [https://sol.sbc.org.br/index.php/sbgames\\_estendido/article/view/19665](https://sol.sbc.org.br/index.php/sbgames_estendido/article/view/19665).
- MERCER, M. **Explorer's Guide to Wildemount**. [S.l.]: Wizards of the Coast, 2020.
- OHNEMUS, A. P. **D & D 5e Mass Combat Simulation**. 2019. Western Oregon University.
- PRESS, K. **Creature Codex for 5th Edition**. [S.l.]: Kobold Press, 2018.
- SHYNE, F. **Automatic Play-testing of Dungeons and Dragons Combat Encounters**. 2023. Honors Theses, Union College. 2745. Disponível em: <https://digitalworks.union.edu/theses/2745>.
- SOMMERVILLE, I. **Software Engineering**. 10 edition. ed. [S.l.]: Pearson, 2015. ISBN 978-0-13-394303-0.
- Wizards RPG Team. **Dungeon Master's Guide**. [S.l.]: Wizards of the Coast, 2014.
- Wizards RPG Team. **Monster Manual**. [S.l.]: Wizards of the Coast, 2014.
- Wizards RPG Team. **Player's Handbook**. [S.l.]: Wizards of the Coast, 2014.
- ZHU, A. et al. Fireball: A dataset of *Dungeons and Dragons* actual-play with structured game state information. In: **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)**. Toronto, Canada: [s.n.], 2023. Disponível em: <https://arxiv.org/abs/2305.01528>.