Automatic Play-testing of Dungeons and Dragons Combat Encounters

By

Fiona Shyne

* * * * * * * * *

Submitted in partial fulfillment

of the requirements for

Honors in the Department of Computer Science

UNION COLLEGE

March, 2023

Abstract

SHYNE, FIONA M. Automatic Play-testing of Dungeons and Dragons Combat Encounters. Department of Computer Science, March, 2023.

ADVISOR: Matt Anderson and TJ Schlueter

Dungeons and Dragons is a game where a player, the Game Master / Game Manager (GM), creates content for a set of other players. It is challenging for GMs to predict the difficulty of potential combat encounters. To aid GMs in balancing combat, we create a simulation environment where virtual agents automatically play-test potential encounters and predict difficulty. We implement several agents to simulate human players that fall into two main categories: rule-based agents that follow a pre-made set of rules and general game-playing agents that explore many potential moves. In simple scenarios, rule-based agents win at a higher rate than general agents, but with complex scenarios, the rule-based and general agents perform similarly. These agents interact in a simulated game environment to play-test potential combat encounters. Our results demonstrate that this simulation outputs similar predictions to from base predictions given from the rule-set of DnD. However, in some situations where our simulation deviated from pre-existing predictions, the predictions from experience GMs align more closely with our simulation than existing systems.

Contents

List of Figures

List of Tables

Figure 1: An example of what an DnD encounter might look like. It uses a grid based system to represent space and physical miniatures to represent creatures.

1 Introduction

We develop an automated tool to assist in the game design of the tabletop role-playing game (TTRPG) Dungeons and Dragons (DnD)[4]. Specifically, we create an automated play-testing system to provide game metrics to estimate challenge of a game. TTRPGs are an interesting test case for this type of tool for three reasons. First, they often have complex and interconnected mechanics that can make predicting outcomes difficult. The second reason is how the role of the *Game Master or Game Manager (GM)*, a player designated with the role of both designing and running games for the other players, impacts the ability to make game content. Unless the GM is running a pre-made module, game material is specific to a table and does not benefit from play-testing prior to the game being played. The last reason is that novice game masters can lack the fundamental understanding of the rule set making it difficult to produce quality content.

Automatic play-testing has the potential to provide more opportunities for GMs in TTRPGs. A part of what makes TTRPGs unique is that much of the story and game content is created the GM, not by the company that designs the game rules. However, the significant effort and knowledge needed by the GM to create game content for their table can often discourage new players from taking on the role of GM. This research hopes to lighten the burden on GMs by providing more robust tools for testing and analyzing potential game content. Automated tools to help GMs can encourage more people to participate in TTRPGs. This is especially important for populations, such as women, that have been historically excluded from

gaming.

In this project, we use the game Dungeons and Dragons 5th Edition (DnD, for brevity) [4] as a test case for how automatic play-testing can benefit TTRPGs. In DnD, an encounter consists of a set of monsters with pre-defined attributes that are controlled by the GM and a set of *player characters (PCs)* with a similar set of attributes that work together to defeat the monsters (see Figure 1). While the official rule books of DnD outline a method to predict challenge of encounters, this method is extremely limited in scope. The goal of our research is to create a proof of concept that predicts challenge of DnD combat encounters.

Our system takes in a potential DnD encounter and simulates games using virtual agents. Then outcomes of the game, such as damage taken or percentage of games where players won, are presented to a GM. We look at both the accuracy of this system, and the usefulness of this system to existing GMs. The accuracy of this system is defined as how close simulated game outcomes correspond to real game outcomes, specifically in regards to game difficulty. In lieu of human play-testing, we use predictions from experienced GMs to estimate this. Usefulness is measured both by how much there is a need for this system, and how much this system would improve DnD games. These concerns are summarized by two research questions:

1. How accurately can automated play-testing systems predict difficulty of DnD combat encounters?

2. How useful are automated tools for testing game balance in DnD to GMs?

Section 3 outlines how DnD encounters are represented in our simulation environment. An encounter describes a set of monsters that fight a set of PCs, called the Party. Creatures compete in turn based combat until one team is completely incapacitate. Our proof of concept simulation represents a sample of the rule-set for combat, with an emphasis on early game combat.

In Section 4 we discuss the different virtual agents we create to make decisions for the creatures in the simulated environment. We test both rule-based agents and general agents. Rule-based agents use a small, predefined rule-set to make decisions. General agents use heuristics and future simulation to estimate which turn is best for the team. In Section 5, we discuss the methods and results for testing how accurate our simulation environment is. We compare our simulation results to both the official guidelines and human graders experienced in DnD. In Section 6, we discuss the methods and results for testing how useful a tool to predict difficulty of encounters would be useful to GMs. We do this through a survey of current GMs of the game.

2 Background and Related Work

Previous work in game studies and computer science has studied how to measure game quality across a variety of games. One way game qualities can be estimated, is through automatic play-testing, which has been shown to be successful across many game types. These techniques have been applied to Role-Playing games similar to DnD.

2.1 Measuring Game Quality

Research in game design has looked at the ways that game quality could be determined and quantified. Liapis et al. [16] explored ways game levels, spanning a wide range of genres, can be evaluated. They concluded the most important factors were symmetry, area control, and exploration. They demonstrated the range of this evaluation by generating maps for both multiplayer strategy and rogue-like games. The usefulness of such quantification has been demonstrated through the creation of new games using these factors as a fitness function. For example, combinatorial games [3] and levels for real-time strategy games [15] have been generated using similar metrics as a fitness function.

One important aspect of game quality is the idea of balance. Game balance describes a variety of concepts, including challenge, fairness, and engagement. For single-player games, balance most often means that the difficulty of the game is proportional to the player's ability. A game that is too hard makes the player frustrated, while a game that is too easy leaves the player unsatisfied. The goal of balancing in this case is to make the game difficult enough to engage the player without discouraging them. For a multiplayer game, balance most often describes that a single play-style in the game should not consistently outperform other strategies. A classic example of a perfectly balanced game is the game of "Rock, Paper, Scissors." For every potential play-style a player could employ, there is a counter-strategy that successfully beats it. This means that no one play-style is consistently more powerful than another over a variety of players. Rock, Paper, Scissors provide a framework that is the basis of many multiplayer games[8]. The goal for balancing a multiplayer game is to make differences between player outcomes a factor of player ability and not chosen play-style of the player, and that players can choose from a variety of possible play-styles and still be successful. Within the context of TTRPGs like DnD, which are cooperative multiplayer games, both factors need to be considered. DnD is a multiplayer game in the sense that each player can employ different play-styles in the game. The clearest example of differing strategies is the choice of character archetypes (species and class) [1] that lead to different play-styles for characters. For example one character

[1] In DnD species and class are aspects of a character that define what skills and abilities they have access to. Species has been historically refered to as race.

may be a fighter and another a healer. Therefore making sure different play-styles are well-balanced and that no race or class is uniformly more powerful than another is important. However, the players all work together to defeat an encounter determined by the game designer, in this case a GM. It is typically desired that an encounter balances difficulty so that the party is challenged by encounters but does not face devastating consequences such as PC death. Within the context of this paper, difficulty is the main measure of balance for an encounter.

2.2 Game Playing AI for Turn-Based Games

Game playing AI has been successful in similar problems to DnD. The General Video Game Playing Competition [19] tested how different algorithms preformed when playing a variety of single-player 2D games. There are several similarities between this competition and combat in DnD. Both DnD and the games from this competition had turn based actions performed on a grid. Additionally, an agent playing DnD must be a capable of making decisions for a variety of creatures, which may require different strategies based on creature characteristics. Algorithms that were successful in this competition had to be capable of several strategies, due to the variety of games it played. Successful agents in this competition use a variety of strategies including variations of Monte Carlo game search, Monte Carlo tree search, and reinforcement learning. Monte Carlo search has also been demonstrated to beat rule-based decision-making for games such as the card game Magic the Gathering [25], which like DnD has a variety of mechanics that vary based on an individual player.

The Nethack competition [14], demonstrated the success of general game playing strategies in a DnD like environment. Similar to the general game-playing competition [19], this is an annual competition that requires participants to develop AI agents that are able to perform well under a variety of situations. The game Nethack, based in part on DnD, is a rich and challenging game. The procedurally generated content of Nethack means that the players have to be prepared for a wide variety of possible challenges. Additionally, there are several classes the player can choose from, which determine how the game can be played. This means an AI agent must be able to perform well with a variety of challenges and a variety of abilities, similar to how a DnD agent must perform. The original competition showed how reinforcement learning and neural networks were successful techniques, however, the most recent competition [12] demonstrates the value of symbolic AI techniques.

2.3 Automatic Play-testing

The time and resources involved in human play-testing makes it infeasible within the context of DnD encounters, created by an individual GM. However, automatic play-testing is a time and resource inexpensive alternative that can provide valuable insight about a game. It has been proposed that effective game-playing AI can be implemented when human play-testing is not feasible [11]. During the development of a game, even in the context of a DnD combat encounter, mechanics and game content change frequently. Even small changes can have unpredictable and large-scale impacts on gameplay. Given that general game agents do not rely on specific game mechanics, they can be used throughout the game development cycle without modification needed to adapt to changing game elements. Automatic play-testing has been shown to successfully identify game quality measures in a wide variety of games including causal mobile game [21], board games [18], card games [7], turn based rogue-likes [13], and shooting games [26].

In DnD, game content is often made for a particular set of players by the GM. This customized game content makes every game of DnD unique and human play-testing infeasible. Additionally, the rule-set of DnD is very complex and game outcomes are unpredictable. Automatic play-testing has been successful in many type of game play and has a strong potential to improve game quality in DnD.

2.4 Evaluation and Game Playing AI for Role Playing Games

Automated tools for game creation and evaluation have been created for other role-playing games. Dahlskog et al. [5] looked at how dungeons, often used in games like DnD, could be formally categorized and analyzed. Defining dungeon structure can be used to generate new dungeons such as in Ashlock et al. [2], where the distance to a goblin's lair was used as a fitness function for creating dungeon maps. Automatic play-testing has also been used to generate and balance encounters in role-playing games. Pfau et al. [20] used deep player behavior modeling to create balanced enemy behavior for the multiplayer role-playing game Aion. However, this required an expansive amount of player data, which would most likely not be possible to collect for a table top game like DnD.

3 Simulation Environment

The vast amount of rules and components in DnD 5e, makes completely recreating a DnD encounter beyond the scope of this project. Therefore we propose a simplified version of DnD, which can be used as an estimate of a complete level 1 encounter for novice players.

In a DnD combat encounter there are three components: a set of monsters, a set of PCs, and a map. The

set of monsters is chosen by the GM, often by selecting them from the Monster Manual [17].These monsters are chosen for a variety of reasons, which may include fitting a narrative or selecting on difficulty. Each PC is chosen or created by a player in the table. Parties may coordinate PC abilities with each other, but they may also act independently. A map describes the location where the combat occurs on. In this simulation, a map is represented as a rectangular grid, where each creature is located on some cell.

The rule set in this simulation environment was determined by modifying the original rules of DnD such that i.) it sufficiently approximates a true DnD combat encounter, ii.) it is simplified enough as to not tax our computational resources. Notably, this environment only simulates low-level encounters (typically done at the beginning of a campaign). A full list of the guiding principles used to create this rule set is listed in Appendix B.

In our simulation environment, creatures (see Figure 2) take turns in the order determined at the start of combat (see Figure 3). Each turn is defined by two elements: a movement and an action, executed in that order. A movement describes a new free grid location within the range determined by the speed of the creature. An action can be chosen from a pre-defined set of actions given to each creature. Actions can accomplish one of two tasks: injuring opponents, or assisting allies. If a creature takes sufficient injury or damage, they become incapacitated and our unable to take turns. Combat ends when an entire party becomes incapacitated.

Our simulation captures a subset of the total available creatures and features of DnD. Pre-defined monsters are available in the Monster Manual ([17]). Our simulation represents all 109 monsters under level 1 from the Monster Manual. However, some monsters have missing or partially implemented components in our simulation environment. PCs are not accessible in the Monster Manual, and therefore we created 14 representatives PCs for this simulation. Spells represent a particular challenge for this simulation, as their rule-set is highly varied. Out of the spells available in level 1, we implement 68 percent (27) of combat-focused spells. Additionally, we include 24 special features (used by both PCs and monsters).

4 Agents

We implement several artificial agents that attempt to model how human players behave. These fall into two general categories: rule-based agents in Section 4.1 and general game-playing agents in Section 4.2. All agents have the goal of defeating opponents and keeping party members alive.

Section 4.3 describes the process for testing the success of agents. Overall, rule-based agents win more games than general game playing agents. However, when choosing actions for more complex creatures general game playing agents have win rates similar to the rule-based agents. This suggests greater com-

BABY

Tiny humanoid, chaotic neutral

Armor Class 12
Hit Points 1 (1d4 - 2)
Speed 30 ft.

STR	DEX	CON	INT	WIS	CHA
7 (-2)	14 (+2)	6 (-2)	4 (-3)	6 (-2)	20 (+5)

Saving Throws Cha +7
Skills Deception +7
Damage Resistances psychic
Condition Immunities charmed
Senses passive Perception 8
Languages —
Challenge 0 (10 XP)

Inexplicable Escape. When grappled Baby can make a DC Dexterity check to escape as it's action.

Spellcasting. The Baby is a 1-level spellcaster. Its spellcasting ability is Charisma (spell save DC 12, +4 to hit with spell attacks). The Baby has the following 3 spells prepared:

Cantrips (at will): *minor illusion*
1st level (2 slots): *command, dissonant whispers*

ACTIONS

Bite. *Melee Weapon Attack:* +0 to hit, reach 5 ft., one target. *Hit:* 1 (1d4 - 3) Piercing damage.

Figure 2: An example of the characteristics of a creature in DnD.

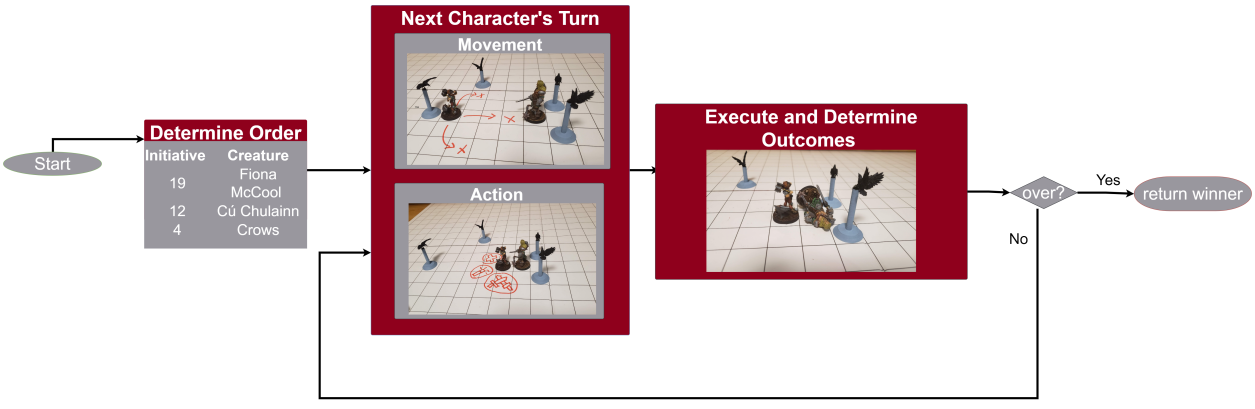

Figure 3: A flowchart of a DnD Encounter. In this example three creatures are engaged in combat. The first creature to act, Fiona, makes a movement decision, and then decided an action. Fiona's action deals enough damage to her opponent to defeat him, ending combat.

plexity within games requires the more nuanced strategies general agents can provide.

4.1 Rule Based Agents

Rule-based agents have knowledge about gameplay in DnD and can exploit that to make decisions. They take in the current game state, and use pre-defined rules to make a decision based on that state. This has the advantage of being fast but is not as flexible as general agents.

All rule-based agents can deconstruct it's available turns into it's movement and action components, and have knowledge of how they interact. A *turn* is both a movement, and an action given as (m, a). While movement is executed before the action, the agent can choose a movement and action in any order. Given a creature c and a game state g, an agent can access the following functions:

- $T(c, g)$: All movement and action combinations that can be made from creature c in game state g.
- $M(c, g)$: All movement that can be made from creature c in game state g.
- $A(c, g)$: All actions that can be made from creature c in game state g.
- $A(m, c, g)$: All actions that can be made from creature c in game state g if movement m is executed.
- $M(a, c, g)$: All movement that can be made from creature c in game state g where action a can be executed following the movement.

A metric the rule-based agents rely on is the estimated damage of an action. If an action can cause damage, the estimated damage is calculated by the average amount of damage it can cause (based on the number and type of dice that are rolled), multiplied by the probability the action will actually cause damage to a particular creature. If an action could effect multiple creatures, this value is multiplied by a constant factor. For all non-damage-causing actions (for example healing) this value is 0.

Out of the five rule-based agents created for this simulation, the most successful (total amount of wins) was the Aggressive Agent (see algorithm 1). The Aggressive Agent first chooses an action that has the highest estimated damage. Once an action is chosen it will choose the movement that is closest to an enemy where this action can be performed. This will only weigh actions that deal damage, so aiding actions such as healing spells are not considered.

4.2 General Game Playing Agents

General game-playing agents do have access to the same knowledge that rule-based agents do. While game mechanics are explicitly referenced in rule-based agents, mechanics have to be "learned" for general-game

Algorithm 1: Rule based agent: Aggressive

Input: A creature C, and a game state G

Choose action a in $A(C, G)$ with the highest estimated damage;

Choose move m in $M(a, C, G)$ that is closest to an enemy;

Return (m, a);

playing algorithms. General agents can access the available turns it can execute, but cannot decompose a turn into it's movement and action components. They also do not have knowledge about damage or other game-play aspects.

The only information they have available to them is a representation of the game state, a list of turns it can make, a list of turns other creatures can make, and a way to create new states by applying actions. In order to determine how well an agent performs in a state a heuristic is also given which outputs how well a particular party is doing. We choose the score of a particular game g for creature c is equal to

$$Score(c, g) = (H_c/M_c) - (H_{\neg c}/M_{\neg c})$$

Where H_c is the sum of the current health for team c belongs to and M_c is the sum of the max health of the team creature c belongs to. The team $\neg c$ is the opponent to the team creature c belongs to.

We test four strategies for general-game playing agents within this simulation. The final general agent chosen took elements from the JinJerry algorithm described from the 2014 General-game playing competition ([19]) and Monte Carlo Game Search ([1]). These algorithms predict future game states by randomly executing turns for each creature, as described in algorithm 2. In DnD, many possible turns are obviously illogical, for example not making any action when multiple are available, therefore we modified concepts from JinJerry and Monte Carlo Game Search to eliminate poor-performing actions in between rounds of future modeling.

Our agent, the Trimming Agent (see algorithm 3), will first execute each turn on a copy of the game state and model a future event only a couple turns in the future. It will then eliminate future models that have low scores. It will then model the remaining turns at an increased depth. It repeats this process over several rounds before returning the turn that has the largest average score across all models.

4.3 Testing Agents

To determine which agent should be used our tool, agents compete in a tournament. For each game, two identical parties are created (sets of creatures), and had one agent control one party and another control the other. Each agent played a set number of games against every other agent. However, due to time constants,

Algorithm 2: Future Model

Input: A game state g, the max depth to simulate to N

Set D to 1 ;
Set s to copy of game state g;
while D is less than N and game s is not over **do**
    Set c to next creature in s ;
    Set t to random turn in $T(c, s)$;
    Execute action t in s;
    Increment D by one ;
end
Return state;

Algorithm 3: General Agent: Trimming Agent

Input: A creature C, A game state G, a list of depths D_0 .. D_n, trimming rate $R \in [0, 1]$.

Set T to $T(C, G)$; `// define T as the set of turns to consider`
for d in D_0 ... D_n **do**
    for Turn t in T **do**
        Set g to copy of G ;
        Execute turn t on g ;
        Set g to FutureModel(g, d);
        Set t_{value} the average of t_{value} and Score(g);
    end
    Remove the $|T| \cdot R$ turns with lowest values from T;
end
Return t in T with highest value;

Agent	Total Wins	Average Clock Time (seconds) Per Turn
Protective Agent	65	0.0007
Aggressive Agent	58	0.0007
Monte Carlo Game Search Agent *	41	4.1837
Trimming Agent *	31	0.5091
JinJerry Agent *	26	0.3437
OLETS Agent *	9	3.3861

Table 1: The total wins and average time in seconds taken per turn for each agent in the all creatures tournament. '*' indicates the agent is a general-game playing agent.

Agent	Total Wins	Average Clock Time (seconds) Per Turn
Monte Carlo Game Search *	54	13.8113
Aggressive	53	0.0016
Protective	44	0.0017
Trimming *	42	2.0536
JinJerry *	37	41.9424
OLETS *	33	5.258

Table 2: The total wins and average time in seconds taken per turn for each agent in the PC tournament '*' indicates the agent is a general-game playing agent.

only a subset of agents are tested in this way. Smaller trials are consistent with the results shown here for all agent sets.

We want to determine the success (win rate) for each agent when playing both monsters and PCs. Therefore we conduct two tournaments. The first includes creatures sets randomly selected from all creatures in simulation, which are predominately monsters. The second tournament includes only PCs. For both tournaments identical creature sets competed against each other, with behavor controlled by opposing agents.

In the tournament with both monsters and PCs, the rule-based agents had a clear advantage over general agents (see Table 1). However, for PCs the agents were more evenly matched (see Table 2). We believe this is due to the greater complexity of PCs compared to monsters, which meant the simplistic strategies of rule-based agents were less successful. However, in all trials, the rule-based agents take significantly less time to make decisions. For these reasons the aggressive agent was chosen as the behavior of simulation.

5 RQ1: How accurately can automated play-testing systems predict difficulty of DnD combat encounters?

Our first research question tests how accurate simulated games are at predicting of an encounter with a particular encounter on a particular set of players. To answer this question we compare the results of simulated games to baseline predictions and human predictors.

5.1 Difficulty Categories

We use difficulty categories as the main measure of the challenge of a encounter. As defined in the DnD Player's Handbook [4], there are 4 levels of difficulty for encounters: Easy, Medium, Hard, and Deadly. The following descriptions are available in the Player's Handbook:

> **Easy**. An easy encounter doesn't tax the characters' resources or put them in serious peril. They might lose a few hit points, but victory is pretty much guaranteed.
>
> **Medium**. A medium encounter usually has one or two scary moments for the players, but the characters should emerge victorious with no casualties. One or more of them might need to use healing resources.
>
> **Hard**. A hard encounter could go badly for the adventurers. Weaker characters might get taken out of the fight, and there's a slim chance that one or more characters might die.
>
> **Deadly**. A deadly encounter could be lethal for one or more PCs. Survival often requires good tactics and quick thinking, and the party risks defeat.

5.2 Methods

To test our simulation we compare a variety of prediction methods, including our simulation, for a sample of generated encounters.

5.2.1 Generating Encounters

A sample of 63 encounters (sets of monsters) was generated, along with two parties (sets of players characters).

Out of the 63 encounters, 14 are taken from adventures written for the game. Fourty encounters were randomly generated (algorithm 4). These encounters were generated from the difficulty predictions discussed in Section 5.2.2, such that each difficulty category had 10 encounters. This was done by first randomly selecting monster challenge, testing if it makes the encounter to hard, and if it doesn't adding a random monster in that challenge. The remaining 12 encounters were designed to test specific hypotheses. Six encounters were generated to test if 5 kobolds would be more difficult than 5 monster of similar challenge. Six encounter were generated to test if a goblin and hobgoblin would be more difficult than 2 monsters of similar challenge. However, neither simulation results nor human grader found significant differences in these encounters, and therefore these hypotheses were dismissed.

Additionally two sets of PCs were created.

Algorithm 4: Random Encounter Generation

Input: A set of monsters $M_0....M_n$, A difficulty category D, number of players p

Set $C_0...C_n$ to challenge ratings in $M_0....M_n$;
Set E to the empty list ;
while *E is not in challenge rating D* **do**
    Set c to random challenge in C ;
    Set m to random monster in M with challenge c ;
    if *adding m to E makes encounter too hard* **then**
        Remove largest challenge in C ;
    else
        Add m to E ;
    end
end
return E

1. **Balanced Party:** The balanced party has a combination of different player archtypes. This is the a kind of party.

2. **Unbalanced Party:** This party contains only one type of PC (magic users). This kind of party is less common in DnD games.

Details about the parties are in Appendix C.

5.2.2 DMG Guidelines for Estimating Difficulty

The Dungeon Master's Guide (DMG) [10] provides a formula for determining the difficulty of an encounter for a given party. The first step is to determine calculate a special value, called the Adjusted XP, for an encounter. The Adjusted XP is calculated as shown in Equation 1. In this formula n represents the number of monsters in the encounter, XP_i represent the challenge value for monster i, and M_n is a static multiplier for an encounter with n monsters. Overall this formula represents the total sum of challenge of individual monsters, adjusted for the added challenge of fighting multiple monsters.

$$A = M_n(\sum_{i=0}^{n} XP_i) \tag{1}$$

The adjusted XP is compared with to thresholds determined by the PCs in the party. For each party 4 thresholds can be calculated by the level of each player in the party, representing the 4 difficulty categories. Whichever threshold is closest to the adjusted XP is the given difficulty of the encounter.

However, there are some ways this system is flawed. The constant multiplier does not take into account emerging difficulty from how specific monsters may interact and the challenge rating does not consider variation amongst monsters at the same level. Additionally the specific strengths or weakness of the PCs

are not considered. Since the two created parties have the same number and level of players, the difficulty predictions for an encounter by the DMG is the same regardless of party.

5.2.3 Simulated Games for Estimating Difficulty

For each sample encounter, we record outcomes from simulated games using the Aggressive Agent as it was the fastest and most effective agent. Each encounter was simulated for 40 games per party (80 times total). Each game produced the following statistics:

- **Total damage**: The amount of damage PCs took in the game divided by the number of PCs.
- **Normalized Damage**: The amount of damage taken by PCs divided by the maximum amount of damage they can take.
- **Success rate**: the number of games won by PCs divided by the number of games.

For each encounter the average and standard deviation values of these statistics from all simulated games were recorded. Encounters are predicted to be harder if damage is high and success rate is low. Table 3 demonstrates how to determine the difficulty based on simulated games results. This is an interpretation of the difficulty categories description from the Players Handbook, discussed in Section 5.1. Generally this was done through modeling scenarios the descriptions implied. For example a medium encounter might have "one or two scary moments," which can occur is PCs get very low on health. For a 5 player party, if 4 PCs take minimal damage (10%), but one PC takes significant damage 80%), the total normalized damage is 0.2. If 2 PCs take significant damage, then normalized damage is 0.4. However, these descriptions are not precise in their language, so there is a wide margin of error for these values.

For Easy encounters, success is "nearly guaranteed" so success rate is set to 1. Additionally PCs should use little resources, so normalized damage rates are fairly low (less then 0.2). For Medium encounter, PC "should" be successful so success rate is set to 1 also. There can be "scary moments" most likely referring to PCs getting to very low HP values. In a 5 player game if one player was reduced to 20% health, while other took minimal damage the normalized damage is likely to fall between 0.2 and 0.4. In hard encounters things "could go badly" meaning success is not guaranteed, but there is only a "slim chance" of a PCs dying. This means success is still more likely than not. Since one or more PCs might die, damage is likely to be higher.

Stat	Easy	Medium	Hard	Deadly
Total Damage	0-2.5	2.6-4.5	4.6-6.5	> 6.6
Normalized Damage	0 - 0.2	0.2-0.4	0.4 - 0.6	> 0.6
Success Rate	1	1	1-0.8	< 0.8

Table 3: A table for translating simulated game outcomes to difficulty category for level 1 players characters.

5.2.4 Predictions from Expert GMs

In situations where simulated games produce different predictions than DMG guidelines, we ask human graders to give their predictions.

We required participants using CloudResearch. They are first asked a series of question that tested their experience and knowledge about DnD as a GM. The participants that passed the screenings reported have the following qualities:

- Played DnD 5e as a player for at least 6 months.
- Played DnD 5e as a GM for at least 3 months.
- Were able to answer a basic question about the game.

After the screenings, participants answer questions about their experience in the game, dicussed in greater detail in Section 6, before being presented with encounters to grade.

In total 4 encounters are presented to each participant. Half of the participants were presented these encounters with the balanced party and half were presented the unbalanced party. The order of encounters are randomly shuffled for each participant. The encounters presented are the following (more details available in Appendix D)

- **Encounter 1**: An encounter with adjusted XP of 250 (medium difficulty) that was predicted to be easy by the simulation.
- **Encounter 2**: An encounter with adjusted XP of 250 (medium difficulty) that was predicted to be hard by the simulation.
- **Encounter 3**: An encounter with adjusted XP of 400 (hard difficulty) that was predicted to be medium difficulty by the simulation.
- **Encounter 4**: An encounter with adjust XP 500 (deadly difficulty) that was predicted to be a medium difficulty by the simulation.

A completed character sheet (e.g. Figure 2) is given for each monster in the encounter. In contrast only a sample of information is presented about the PCs in the party (Species, Class, Armor Class, and Hit Points).

A total of 328 participants opened the survey, of which 69 successfully completed screening measures. Out of the 69 participants, 46 were men and 23 were women. The racial breakdown of our sample included 44 White individuals, 15 Black individuals, 5 Asian individuals, 5 Lantinx individuals, and 1 mixed race individual. The median age was 32, with the youngest participant being 21 and the oldest being 55. Additionally, 58 participants identify as straight, 9 identify as bisexual, and 2 identify as gay.

5.3 Results

Simulated games produce similar predictions to the predictions given by the DMG, but there are many instances where simulation result differ from DMG predictions. Predictions from GMs, in at least some cases, align more with simulated predictions than with DMG predictions.

5.3.1 Simulation Outcomes

The difficulty predictions from the DMG correlate strongly with simulated game outputs. However, there is large variation of the simulated outcomes within each difficulty category predicted by the DMG. This variation is caused both by PC and monster composition. In the Balanced Party, PCs take significantly less damage than the Unbalanced Party. Further, in Encounter 1 PCs took less damage than in Encounter 2, despite both encounters being predicted to be equally challenging in the DMG.

General Results

The difficulty predictions from the DMG has a significant influence on the outcomes of simulated games Figure 4. A one-way Anova on total damage by DMG difficulty category shows significant results ($F(4, 63) = 19.52$, $p < 0.001$). Post-hoc tests shows significant difference between all categories apart from Easy and Medium as shown in Table 4.

Appendix F shows the simulation results broken down by adjusted XP. While this also shows a positive correlation between DMG metrics and simulation results, it demonstrates that that there is a lot of variability of simulated outcomes even within one adjusted XP value. These is influenced by both PC composition and monster composition.

PC Composition

In Section 5.2.1 two parties are described: the Balanced Party and the Unbalanced Party, where the Balanced Party is a combination of magic and non-magic users and the Unbalanced Party is all magic users. Figure 5 shows the average damage taken for each party. Despite the DMG predicting both parties to be equally equipped, the Balanced Party takes less damage than the Unbalanced Party across all DMG predicted difficulties. An Independent T-test shows that the Balanced Party takes significantly less damage

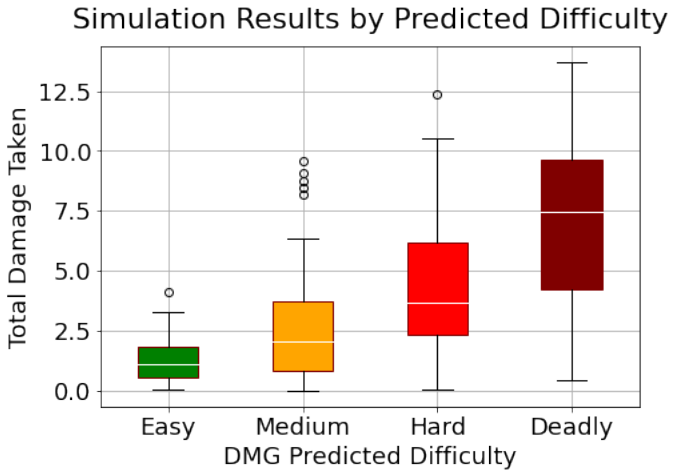

Figure 4: Average total damage from simulated by DMG predicted difficulty categories discussed in Section 5.2.2

Category 1	Category 2	Mean Difference	P-Value
Easy	Medium	1.4364	0.1251
Easy	Hard	3.2394	0.001*
Easy	Deadly	5.5424	0.001*
Medium	Hard	1.803	0.0316*
Medium	Deadly	4.1061	0.001*
Hard	Deadly	2.3031	0.0224*

Table 4: Results of Tukey-Tests for average total damage group by DMG predicted difficulty category. The asterisk (*) represents significance.

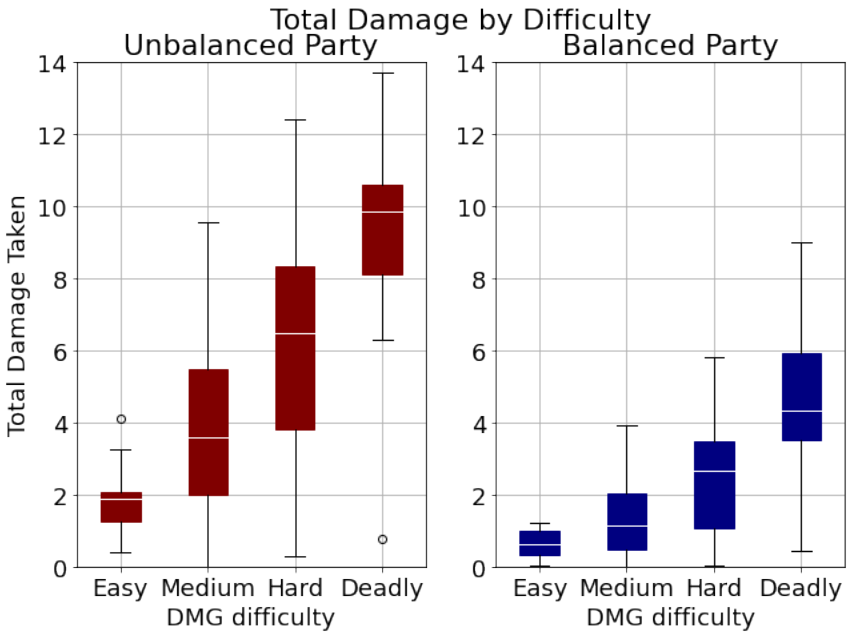

Figure 5: Average total damage taken in encounter played by the Unbalanced Party (left) or the Balanced Party (rights). Results are grouped by difficulty categories predicted by the DMG.

Encounter	DMG Prediction	Simulation Prediction	Adjusted XP	Average Total Damage	Success Rate
Encounter 1	Medium	Easy	250	1.432	1
Encounter 2	Medium	Hard	250	6.34	0.9
Encounter 3	Hard	Medium	400	2.765	1
Encounter 4	Deadly	Medium	500	4.418	0.99

Table 5: Results of Tukey-Tests for average total damage group by DMG predicted difficulty category. The asterisk (*) represents significance

the the Unbalanced Party (t(63) = 5.6423, $p < 0.0001$). Our simulation predicts that encounters are easier for the Balanced Party than for the Unbalanced Party. These results are replicated when the general agent is used, discussed in Section E.2.

Monster Composition

Even taking PC composition into account, our simulation predicts certain monster compositions to be easier or harder than the DMG predicts. Four of the most divergent were sampled to be presented to human GMs. The DMG metrics and simulation outputs are presented in Table 5.

5.3.2 Predictions of GMs

Experienced GMs are presented with 4 encounters, either played by the Balanced Party or the Unbalanced Party. For each monster and party pair the GMs give the encounter a difficulty category (easy to deadly).

Group1 (Encounter / Party)	Group2 (Encounter / Party)	Mean Difference	P Value
2 / Balanced	4 / Unbalanced	0.7429	0.0124
2 / Unbalanced	3 / Balanced	-0.8496	0.0020
2 / Unbalanced	1 / Balanced	-0.7924	0.0055
4 / Unbalanced	3 / Balanced	-1.1143	< 0.0001
4 / Unbalanced	3 / Unbalanced	-0.8529	0.0021
4 / Unbalanced	1 / Balanced	-1.0571	< 0.0001

Table 6: All significant pair differences for GM difficulty predictions across encounter / party groups found by Tukey test. All insignificant pairs were omitted

The responses are given in detail in Appendix F. For analysis these categories were given numerical values (0-3). An Anova of these values grouped by encounter / party pairs showed significant results ($F = 6.558$, $p = < 0.0001$). The post-hoc tests are shown in Table 6.

The most interesting point of comparison is the predictions from GMS of Encounter 1 and Encounter 2 with either the Balanced Party or the Unbalanced Party. All four monster / party categories are predicted by the DMG to be medium difficulty with an adjusted XP of 250. However, our simulation predicts the encounters with the Balanced Party to be lower difficulty than the Unbalanced Party, and Encounter 1 to be lower difficulty than Encounter 2. Figure 6 shows the mean prediction value for these four groups. Similar to simulation predictions, the GMs found Encounter 1 with the Balanced Party to be easier than Encounter 2 with the Unbalanced Party. Encounter 1 with the Balanced Party had a mean prediction value of 0.943 (medium) and Encounter 2 with the Unbalanced Party has a mean category value of 1.73 (hard).

Generally for Encounters 3 and 4, the mean category value from GM predictions can be rounded to 1 (medium). The one exception is Encounter 4 for the Balanced Party where the mean category value was 2.0 (hard). This generally aligns with the simulation predictions, and disagrees with DMG predictions.

Overall mean difference between encounter / pair groups of the GM predictions align with simulation predictions, but not all difference were found to be statistically different. This could be due to the wide variability of predictions from GMs. The average standard deviation of category values was 0.876 across the 8 encounter / party pairs, which is almost a complete category difference. Even amongst experienced GMs, there was disagreement about the difficulty of encounters. This demonstrates the complexity of the task at hand.

5.4 Discussion

Across many encounters, the predictions from the base rule-set has an effect on outcomes of simulated games. However, there are several situations where simulated outcomes do not match DMG predictions.

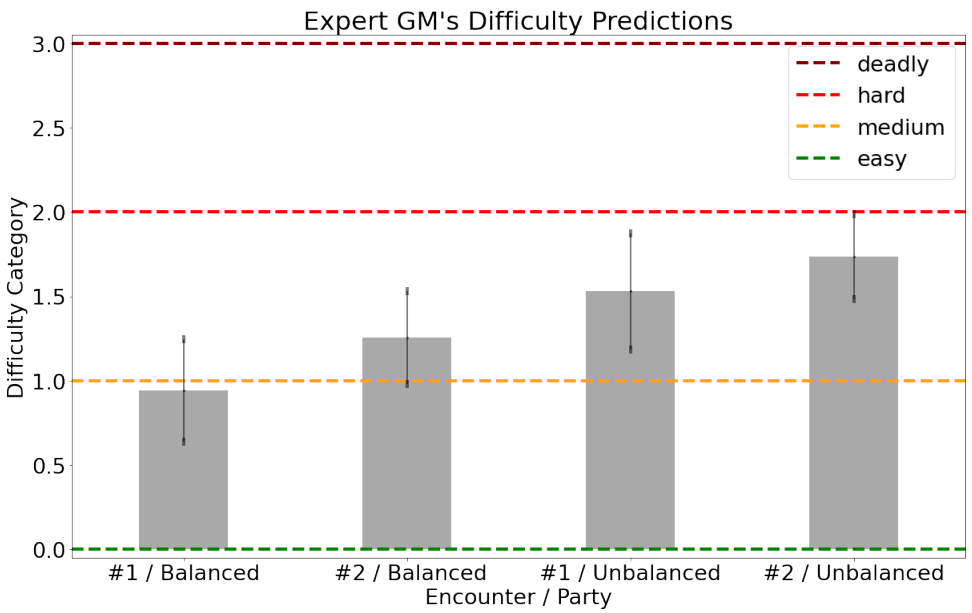

Figure 6: GM predictions for the 4 encounter / party pairs that were found to be medium difficulty by the GM.

These deviations are influenced by both by the PCs and the monsters. Using a sample of these deviations, we asked experienced GMs to make predictions. Overall GMs agreed with predictions made by simulated games over predictions made by the DM. This demonstrates the usefulness of our simulation environment as a proof of concept for how difficulty can be better predicted in the game DnD.

6 RQ2: How useful are automated tools for testing game balance in DnD to GMs?

We survived current GMs about their experienced with DnD to determine the need and desire for tools to evaluate combat encounters.

6.1 Methods

We survey current GMs as described in Section 5.2.4. Before participants evaluated encounters, they were asked questions about their experience with DnD that were a combination of numeric and text based answers. Table 7 provides a sample of the questions that were asked.

#	Question	Format
1	What aspects of DnD do you focus on most in your games?	Text
2	Approximately what percent of sessions are focused on combat?	Scale 0 - 100
3	Approximately what percent of your time spent planning sessions are focused on combat?	Scale 0-100
4	If you use the the Dungeon Masters Guide (DMG) guidelines, how accurately do they predict difficulty of combat encounters in your games?	Text
5	How often do the balance of encounters negatively affect the games you DM or play in?	Scale - 100
6	What effect do the balance of encounters have on the games you DM for or play in?	Text
7	If there was a tool available to you that predicted the difficulty of encounters, how likely would you be to use it?	Text

Table 7: Questions GMs were asked and the type of response that was recorded. Note that DM refers to GM in the context of DnD.

6.2 Results

GMs spend a significant amount of planning and gameplay dedicated to combat. GMs have a wide variety of experience with the available tools for testing combat, with many GMs agreeing the DMG is at least partially inaccurate. The majority of DMs would be open to using new tools to test combat.

6.2.1 Time Spent in Combat

When asked about what aspects of the game they focused on (Question 1 in Table 7), 35 (50.7%) GMs mention role-playing or world-building where only 13 (18.8%) mention combat. The significance of world building is exemplified by one GM's response.

> *I focus mostly on the character interaction aspect of it. The roleplaying and worldbuilding. The combat is of course a staple, but [world-building] is much more interesting!*

However, GM's claim they spend over half of both gameplay (mean = 51.84%, std = 19.05) and planning time (mean = 51.77%, std = 20.022) dedicated to combat (Question 2 and 3 in Table 7).

6.2.2 Accuracy of DMG Predictions

Question 4 in Table 7 asks how accurate GMs perceive the DMG guidelines for difficulty predictions. The text responses from these questions were labeled by researchers on a 1 (not at all accurate) to 5 (extremely

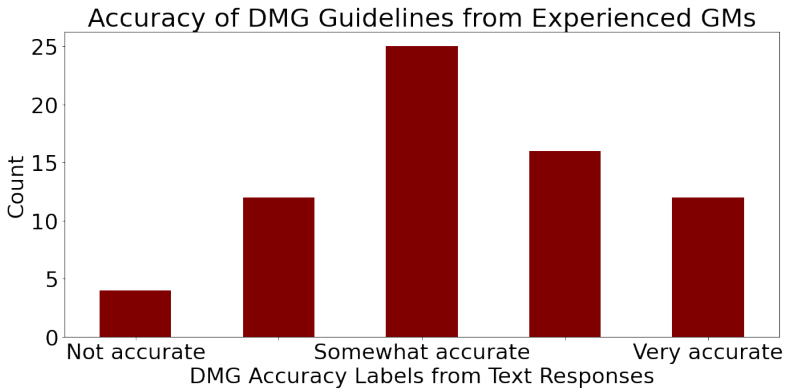

Figure 7: GM's answers of question 4 in Table 7 about the accuracy of the DMG in predicting difficulty. Text responses are labeled by researchers on a scale from 1 to 5.

accurate) scale. Figure 7 shows there was a wide range of perception of the DMG's method for predicting difficulty, with around 28(40.6%) answers being positive, 25 (36.2%) answers being neutral, 16 (23.2%) responses being negative.

The text responses gives additional insight of GMs perception of the DMG. One GM responded that it does not "take into account action economy," referring to the amount and types of action creatures can use in a turn. Several GM's mention they only use the DMG as a baseline and *"fudging things in one direction or another,"* based on their personal experience. Some GMs also mention player skill, one saying the DMG underestimates difficulty due to their *"group's incompetence"* and another saying the DMG overestimates difficulty as their players are *"more advanced than the average player."*

6.2.3 Effect and Frequency of Unbalanced Encounters

Question 5 in Table 7 asks how often encounters are negatively impacted by balance (Figure 8). GMs say on average 32.86% (std = 23.88) of encounters are negatively impacted by balance. 17 (24.63%) GMs say over half of encounters are negatively effect and 2 (2.90%) GMs say over 90% of combat is negatively effected. GMs have revealed that poorly balanced encounters can kill moral, make players board, or derail story lines (Question 6 in Table 7).

6.3 GM Use of Automated Tools

Question 7 in Table 7 asks GMs how likely they would be to use a tool that predicted the difficulty of combat encounters. 56 (81.15%) GMs say they would at least try it, with 25 (36.23%) GMs saying they would be extremely likely to use it.

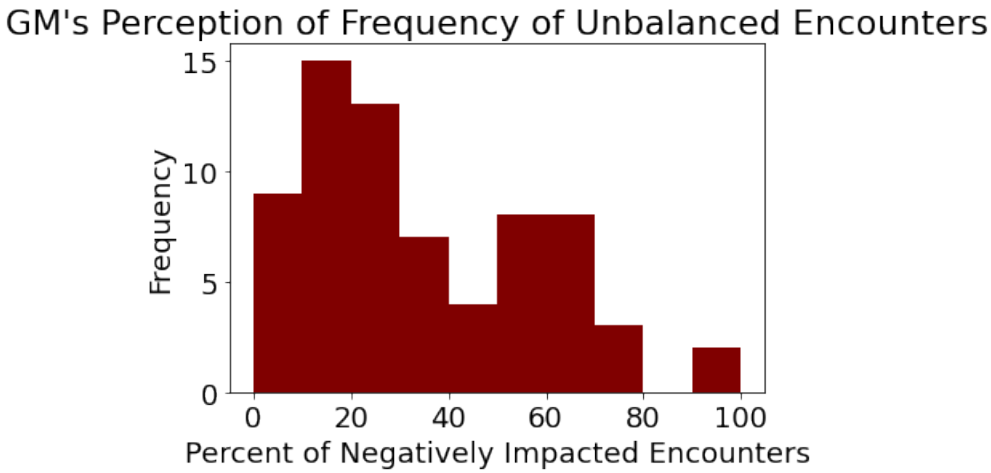

Figure 8: GM's perception how frequently Encounters are negatively impacted by Balance

6.4 Discussion

While GMs are more likely focused on role-play and world-building, they spend a significant amount of time planning and running combat. For many GMs the given system for predicting difficulty is not accurate, and over 30% of encounters are negatively impacted by poor balance. Unbalanced games have the potential to negatively impact the players and their enjoyment of the game. Additionally, GMs overwhelmingly welcome using new systems for predicting difficulty.

7 Conclusion

As a proof of concept, we create a system that automatically play-tests DnD combat encounters and presents the outcomes to the GM. Even with a sample of the total rule-set, our simulation found insight about the game. Over many encounters, simulated games show a correlation with predictions from the official guidelines. However, in a sample of encounters where simulation results different from guidelines, experienced GMs align closer to our simulation predictions than guidelines.

There is a desire from GMs for better tools for predicting accuracy of combat encounters. GMs have a wide variety of perception of the current system of prediction, and the vast majority would use automated tools if given the chance.

There are many limitations of this work. We use human predictions as the main source of validation of our system. However, there is no guarantee human estimates are more accurate than other systems. Future work can explore using human play-testing as another method of validation.

Another significant limitation is the small scope of our simulation environment. By limiting the en-

vironment to rule available in low-levels, there is no guarantee this would be as successful later in DnD campaigns. Future work can look at expanding this environment. An interesting avenue to explore is other TTRPG system beyond DnD. The same principles explored in this paper can potentially be applied to a wide range of role-playing games.

Future work can also explore a greater variety of agent behavior. One element of difficulty that our GMs discussed was player skill. Players with more experience are better equipped to handle encounters that novice players may struggle with. Play-style is another factor that can greatly impact the outcomes of games. Expanding the agents to better reflect how individual players would behave would improve the accuracy of simulated games.

Automated systems have the potential to encourage new players to gain confidence to lead games. Steep learning curves, complex rule-sets, and negative combat outcomes can discourage new GMs from continuing the game. This is especially important to consider within a gaming culture that is often hostile to women [24], people of color [9], and people in the LGBTQ+ community [22, 6] It should not be ignored than even in our small sample, our participants are predominately straight, white men. While TTRPGs can be a great resource for identity exploration [23], many identities have been excluded from it. Our system provides a tool that does not rely on intuition or experience, which can lower the barrier of entry and create better starting experience to new GMs. Our hope is can encourage populations that have been historically excluded from gaming to become GMs and continue in the game. This can allow for a greater number of stories to be told, from a greater pool of creators.

This research can extend beyond the area of TTRPGs. There are many similarities between the mechanics of TTRPGs and other genres of games. It is not difficult to imagine how a tool to analyze DnD encounters could be applied to games with similar turn-based fighting mechanics. Making seemingly minor changes to game mechanics can have major effects on gameplay, which could require expensive and time consuming play-testing to discover. Outside of game development, many areas could benefit from a greater diversity of human input including science, public policy, and engineering. The vast amount of technical expertise, however, hinders many people from contributing to these fields. By using computation to offload some of this technical labor, people with a variety of skill sets could contribute to a wider range of endeavors.

References

[1] Bruce Abramson. “Expected-outcome: A general model of static evaluation”. In: *IEEE transactions on pattern analysis and machine intelligence* 12.2 (1990), pp. 182–193.

[2] Daniel Ashlock and Cameron McGuinness. "Automatic generation of fantasy role-playing modules". In: *2014 IEEE Conference on Computational Intelligence and Games*. IEEE. 2014, pp. 1–8.

[3] Cameron Browne and Frederic Maire. "Evolutionary game design". In: *IEEE Transactions on Computational Intelligence and AI in Games* 2.1 (2010), pp. 1–16.

[4] Jeremy Crawford et al. *Player's handbook*. Wizards of the Coast LLC, 2014.

[5] Steve Dahlskog, Staffan Björk, and Julian Togelius. "Patterns, dungeons and generators". In: *Foundations of Digital Games Conference, FDG, Pacific Grove, USA (2015)*. Foundations of Digital Games. 2015.

[6] Desirée Elveljung. "The use of homophobic pejoratives among gamers A critical discourse analysis of slurs within the gaming sphere". In: (2018).

[7] Pablo Garcıa-Sánchez et al. "Automated playtesting in collectible card games using evolutionary algorithms: A case study in Hearthstone". In: *Knowledge-Based Systems* 153 (2018), pp. 133–146.

[8] Yotam I Gingold. "From Rock, Paper, Scissors to Street Fighter II: Proof by construction". In: *Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames*. 2006, pp. 155–158.

[9] Kishonna L Gray. "Deviant bodies, stigmatized identities, and racist acts: Examining the experiences of African-American gamers in Xbox Live". In: *New Review of Hypermedia and Multimedia* 18.4 (2012), pp. 261–276.

[10] Scott Fitzgerald Gray et al. *Dungeon master's guide*. Wizards of the Coast, 2014.

[11] Cristina Guerrero-Romero, Simon M Lucas, and Diego Perez-Liebana. "Using a team of general AI algorithms to assist game design and testing". In: *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE. 2018, pp. 1–8.

[12] Eric Hambro et al. "Insights From the NeurIPS 2021 NetHack Challenge". In: *arXiv preprint arXiv:2203.11889* (2022).

[13] Christoffer Holmgård et al. "Automated playtesting with procedural personas through MCTS with evolved heuristics". In: *IEEE Transactions on Games* 11.4 (2018), pp. 352–362.

[14] Heinrich Küttler et al. "The NetHack learning environment". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7671–7684.

[15] Raúl Lara-Cabrera, Carlos Cotta, and Antonio J Fernández-Leiva. "On balance and dynamism in procedural content generation with self-adaptive evolutionary algorithms". In: *Natural Computing* 13.2 (2014), pp. 157–168.

[16] Antonios Liapis, Georgios Yannakakis, and Julian Togelius. "Towards a generic method of evaluating game levels". In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Vol. 9. 1. 2013, pp. 30–36.

[17] Mike Mearls and Jeremy Crawford. *Monster Manual: Dungeons and Dragons*. Wizards of the Coast, 2014.

[18] Fernando de Mesentier Silva et al. "AI-based playtesting of contemporary board games". In: *Proceedings of the 12th International Conference on the Foundations of Digital Games*. 2017, pp. 1–10.

[19] Diego Perez-Liebana et al. "The 2014 General Video game Playing Competition". In: *IEEE Transactions on Computational Intelligence and AI in Games* 8.3 (2015), pp. 229–243.

[20] Johannes Pfau et al. "Dungeons & replicants: automated game balancing via deep player behavior modeling". In: *2020 IEEE Conference on Games (CoG)*. IEEE. 2020, pp. 431–438.

[21] Edward J Powley et al. "Semi-automated level design via auto-playtesting for handheld casual game creation". In: *2016 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE. 2016, pp. 1–8.

[22] Bonnie Ruberg. "The precarious labor of queer indie game-making: Who benefits from making video games "better"?" In: *Television & New Media* 20.8 (2019), pp. 778–788.

[23] Toriana Shepherd. *Roll for Identity: A Study of Tabletop Roleplaying Games and Exploring Identity*. University of Wyoming, 2021.

[24] Wai Yen Tang and Jesse Fox. "Men's harassment behavior in online video games: Personality traits and game factors". In: *Aggressive behavior* 42.6 (2016), pp. 513–521.

[25] Colin D Ward and Peter I Cowling. "Monte Carlo Search Applied to Card selection in Magic: The Gathering". In: *2009 IEEE Symposium on Computational Intelligence and Games*. IEEE. 2009, pp. 9–16.

[26] Alexander Zook, Eric Fruchter, and Mark O Riedl. "Automatic playtesting for game parameter tuning via active learning". In: *arXiv preprint arXiv:1908.01417* (2019).

Appendices

A Terms

Acronym	Meaning	Definition
DnD	Dungeons and Dragons	A popular TTRPG game.
DMG	Dungeon Master's Guide	A main rule-set for DnD, geared towards GMs.
~	Encounter	The set of monsters in a game.
GM	Game Master / Game Manager	A special player who creates game content.
~	Party	The set of PCs in a game.
PC	Player Character	The character players control.
TTRPGs	Table Top Role Playing Games	A category of games where one player will create game content.
XP	Experience Points	A challenge value to a monster.

B Guiding Principles

In order to determine how to simplify a DnD encounter for this simulation we develop a series of guiding principles. The base rule set of DnD is reduced using these principles.

- Players and monsters should select from the fewest number of choices possible.
  - Justification: New players and GMs alike don't want to be overloaded with possible actions when they are just learning the game.
  - Application: There are no bonus actions in this simulation.
- Rules and mechanics can be condensed to what would happen most of the time, even if there are some outlier cases.
  - Justification: Rules or mechanics that are infrequently used are less likely to have major impacts on simulation results.
  - Example: Opportunity attacks (a creature's ability to attack a fleeing enemy) will occur at the first time they could happen, as opposed to someone waiting for a better opporuntity
- Game content available to the simulation, such as special features or spells, can be limited to the most frequent instances or categories.

  - Justification: If a game object is not frequently used, it is less likely to have a significant effect on a simulation result.
  - Example: The petrified condition is not implemented.

- This simulation is limited to players at level 1 and monsters at challenge level 1
  - Justification: This project is necessarily small in scope, and the number of rules and caveats after level 1 grows significantly
  - Example: Shapeshifting for druids is not implemented

- For mechanics that have large action spaces, where a creature has to choose from a large set of options, the simulation can use a random sample of those spaces rather than the complete action space.
  - Justification: A random sample can be used as an estimate for the power of the base mechanic.
  - example: The possible movement a creature can make is reduced by a factor of 2 in this simulation.

- Rules and mechanics that are not predominantly focused on combat are not considered in this simulation.
  - Justification: These mechanics are expected to have a small effect on the outcomes of combat.
  - Example: The heart sight spell (knowing the emotions of a creature) is not implemented in this simulation.

C Parties

C.1 Unbalanced Party

The Unbalanced Party consists of the following PCs, all are level 1.

- An Elf wizard with an Armor Class of 12 and 13 hit points
- A Gnome Sorcerer with an Armor Class of 15 and 9 hit points
- An Elf Wizard with an Armor Class of 12 and 8 hit points
- An Half-Orc Warlock with an Armor Class of 15 and 10 hit points
- An Elf Druid with an Armor Class of 14 and 10 hit points

C.2 Balanced Party

The Balanced Party consists of the following PCs, all are level 1.

- A Dwarf Cleric with an Armor Class of 18 and 11 hit points
- An Elf Wizard with an Armor Class of 12 and 8 hit points
- An Halfing Rogue with an Armor Class of 14 and 9 hit points
- A Human Fighter with an Armor Class of 17 and 12 hit points
- A Human Fighter with an Armor class of 14 and a 12 hit points

D Encounters

D.1 Encounter 1

Encounter 1 has the following monsters:

- 1 Stirge with Challenge Rating $1/8$, Armor Class of 14, and 2 Hit Points
- 2 Swarm of Bats each with Challenge Rating $1/4$, Armor Class of 12, and 22 Hit Points

D.2 Encounter 2

Encounter 2 has the following monsters:

- 1 Kobold with Challenge Rating $1/8$, Armor Class of 12, and 5 Hit Points
- 1 Giant Weasel with Challenge Rating $1/8$, Armor Class of 13, and 9 Hit Points
- 1 Guard with Challenge Rating $1/8$, Armor Class of 16, and 11 Hit Points
- 1 Camel with Challenge Rating $1/8$, Armor Class of 9, and 15 Hit Points
- 1 Giant Rat with Challenge Rating $1/8$, Armor Class of 12, and 7 Hit Points

D.3 Encounter 3

Encounter 3 has the following monsters:

- 1 Brown Bear with Challenge Rating 1, Armor Class of 11, and 34 Hit Points

D.4 Encounter 4

Encounter 4 has the following monsters:

- 1 Swarm of Wasps with Challenge Rating $1/2$, Armor Class of 12, and 22 Hit Points
- 1 Dretch with Challenge Rating $1/4$, Armor Class of 11, and 18 Hit Points
- 1 Mastiff with Challenge Rating $1/8$, Armor Class of 12, and 5 Hit Points
- 1 Kobold with Challenge Rating $1/8$, Armor Class of 12, and 5 Hit Points
- 1 Acolyte with Challenge Rating $1/4$, Armor Class of 10, and 9 Hit Points

E Supplementary Results

Apart from the major findings discussed in Section 5.3.1 and Section 5.3.2, some interesting results came out simulations.

E.1 Standard Deviation

Apart from pure difficulty metrics, our simulation capture variability measures. From this GMs could estimate not only the game outcomes, but the expected range of outcomes. If the standard deviation is high for an encounter, the GM should be prepared for a wide variety of potential outcomes.

We find that both DMG predicted difficulty and party composition have an effect on the standard deviation of total damage taken. The Balanced Party had smaller standard deviation values than the Unbalanced Party, as shown in Figure 9. Similarly the standard deviation increased as DMG predicted difficulty increase, as seen in Figure 10.

E.2 Simulated Games by Agent Type

We run the same encounter set with both the rule-based agent (Aggressive) and the general agent (Trimming), as seen in Figure 11. We find that the trends are similar across agent types, however with the general agent the PCs are more successful, making easier encounters. This is most likely due to the fact that the Trimming agent did better in the tournament with just PCs then in the tournament with monsters.

We also look at difference between parties for simulations using the general agent, shown in Figure 12. Similar to the rule-based agent, the Balanced Party takes significantly less damage the the Unbalanced Party

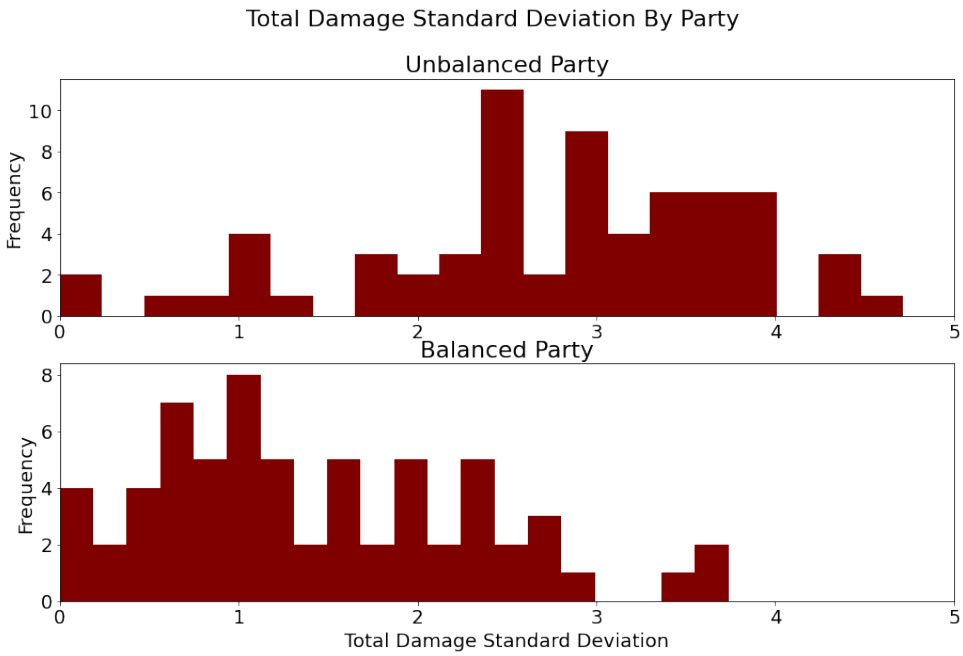

Figure 9: Histogram of Total Damage Standard Deviation across Parities

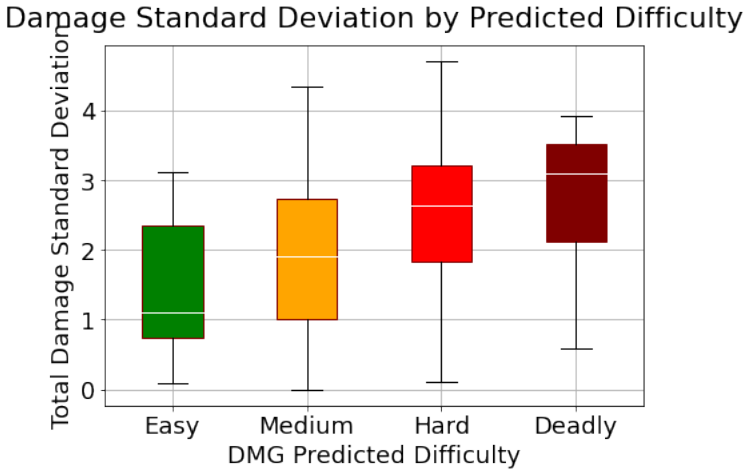

Figure 10: Total Damage Standard Deviation across DMG predicted difficulty

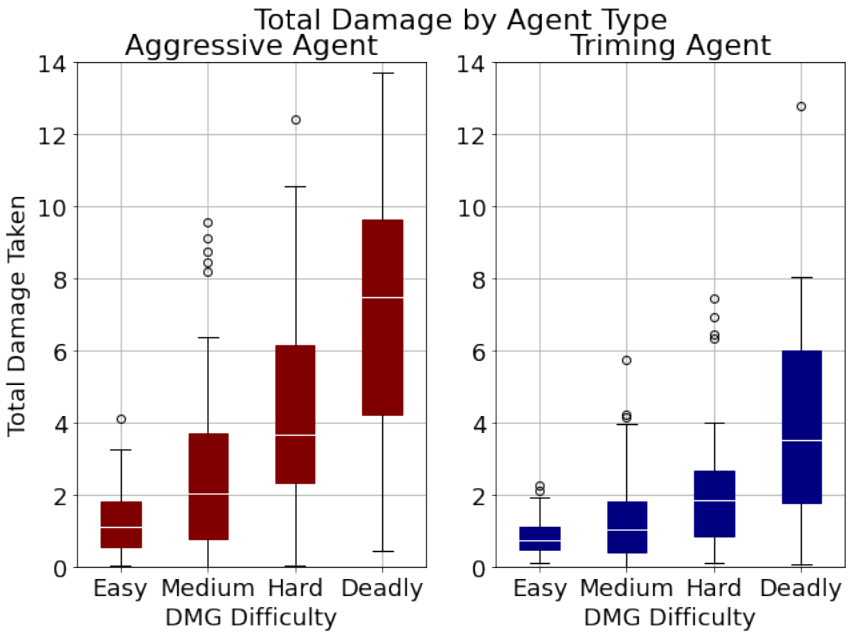

Figure 11: Total Damage Taken by Agent Type

across all encounters. This is confirmed with an independent t-test of total damage ($t = 4.730$, $p < 0.0001$), which shows significant results.

F Supplementary Figures

Figure 13 show the total damage from simulated games by the adjusted XP value of the encounter. Figure 14 shows the predictions made by GMs for Encounter 1 and Encounter 2. Figure 15 shows the predictions made by GMs for Encounter 3 and Encounter 4.

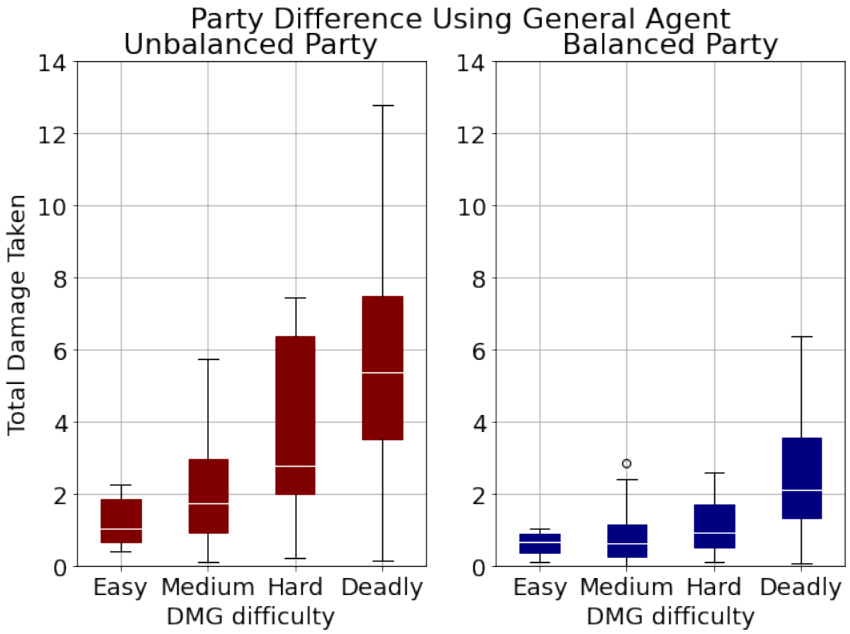

Figure 12: Average total damage taken in encounter played by the Unbalanced Party (left) or the Balanced Party (rights) for simulations using the general agent. Results are grouped by difficulty categories predicted by the DMG

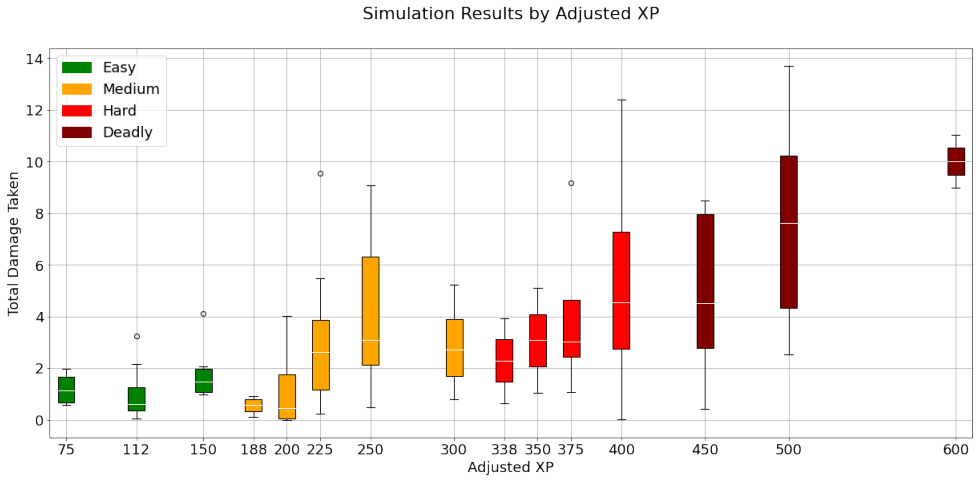

Figure 13: Average total damage from simulated by adjusted XP using Equation 1. Colors represent the difficulty category of that Adjusted XP value. Note certain adjusted XP values had more encounter in the sample then others.

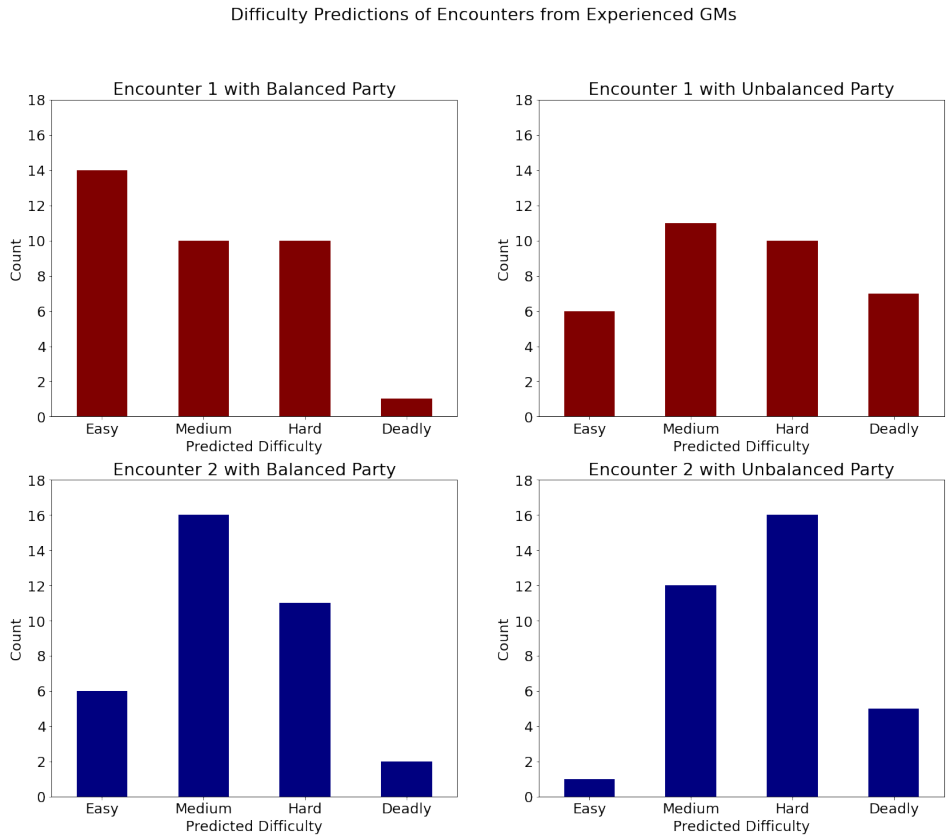

Figure 14: Histogram of GM difficulty predictions for Encounter 1 and Encounter 2

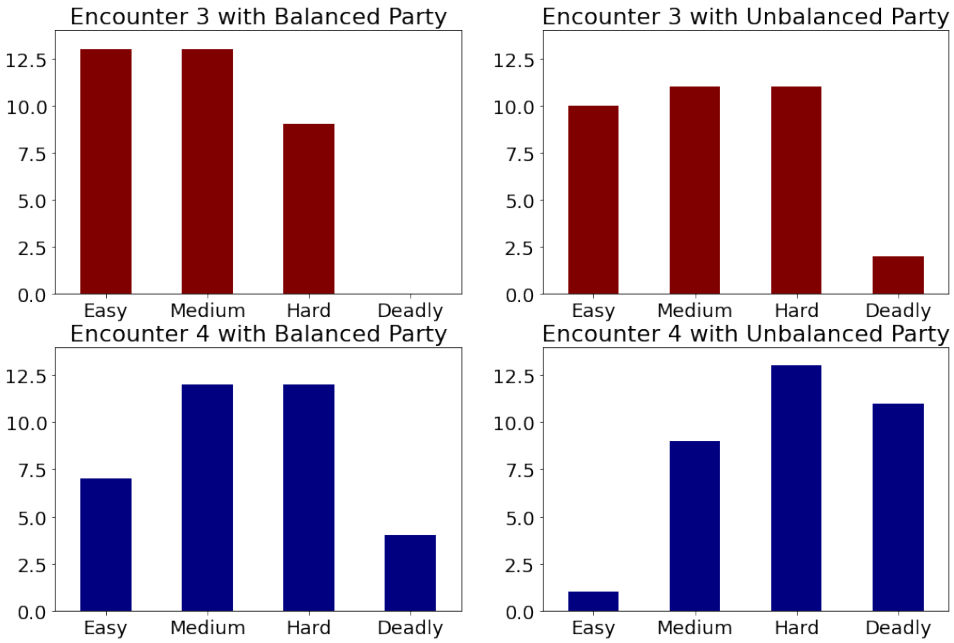

Figure 15: Histogram of GM difficulty predictions for Encounter 3 and Encounter 4