Dungeon Master Battle Duels

Jacob Spigle, Zachary Painter, David Akridge, Hunter Figueroa

October 24, 2017

Contents

Introduction

Role-playing games are games where players act as their characters. Usually in person, around a table, and controlled by the players speaking their actions in turn. This might not always be done by speaking in-character, but rather by following this basic structure:

1. The GM describes the environment.

2. The players describe what they want to do.

3. The GM narrates the results of the characters? actions.

There are millions of possible character, monster, and scenario possibilities when constructing a tabletop role-playing campaign, however, it is hard and sometimes impossible to sift through them all and find the perfect encounter. Not only that, but once you commit to a game or role-playing decision you are forced to see it through to completion in order to retain the game?s continuity. Players get one chance to design their character at the very start of a session, and once they do, that character?s path cannot be meaningfully changed. Spending so much time working on a character, and then coming to the first session to find out that they are severely unprepared for the tasks they are presented can be harrowing. On the other side, the GM spends so much time outside of the game to prepare a story and encounters that challenge the Players, so when that GM accidentally wipes out the entire team of Player Characters (PCs) in one unbalanced encounter, or if the PCs simply walk through a fight that was meant to challenge them for the remainder of the session, the GM feels they have failed the PCs in presenting a good game. This project presents a service that can be used by both GMs and PCs alike to test out their characters, encounters, team builds, or boss fights in a simulated environment. This will allow users to figure out just how well their creation meets the requirements they have set for themselves. The initial scope of this project will seek to develop a service that takes the rules and gameplay of the Dungeon and Dragons 5th Edition D20 System, and allows PC or GM to import their character's/monsters' statistics and equipment, then play against an automated opponent. To further enhance a user?s ability to test and refine encounters and PCs, users will be able to upload and share their encounters for others to try.

Proposed Solution

Broader Impact

Tabletop games bring people together in a world where physical interaction is often lost in lieu of digital communication. This project aims to supplement these interactions by making them more accessible to newer players, as well as help experienced players spend less time on pre-game preparation. This project has the potential to broaden the audience of an already quickly expanding pastime to younger players and DM's. Concepts in the realm of tabletop games are very easily understood by a younger audience, as their imagination and creativity can run free in such games. However, the threshold of understanding for the many rules and balances may prove a bit overzealous for this audience. Having a service that allows improves accessibility to newer players will also bleed into improving accessibility for younger players as well. Additionally, users who do not have a group to play with may find and form groups with other users with whom they have played and shared content with.

Personal Interests

4.1 Jacob Spigle

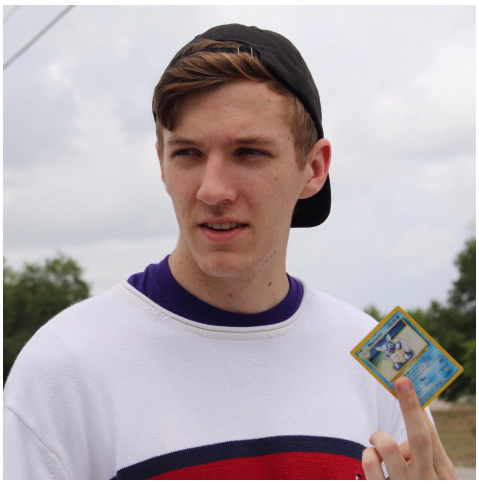
I discovered role-playing games like Dungeons & Dragons early this year. After playing video games since childhood, D&D was a breath of fresh air; a way to mix the excitement of creating a character, leveling up, deciding your actions, along with an openness that allows nearly infinite stories, infinite worlds, and the tools to create those stories and worlds yourself. Those infinite possibilities can spur creations that might not work as well ingame as you thought. I have written campaigns and characters that on paper seemed great, but when brought to the table were found lacking. When I searched through the many different forums and groups I had joined after I started playing D&D, and scouring the internet for a way to playtest my creations, I was surprised to found that no such tool existed. This service is something I can actually use and wish that I had during the creation of campaigns I have written, and that is why I believe others will find use in it as well.

4.2 Zachary Painter

Some of my primary interests in Computer Science are data structures and algorithms. In my area of research, concurrent programming, these two topics are frequently discussed. Often times in this field, I have had the opportunity to explore many complicated and fascinating algorithms. When tackling these kinds of problems, you often find yourself spending considerably more time thinking and reasoning than implementing/coding. After understanding a problem, you move on to another without getting the chance to apply what you learned in a practical/useable way. This constant cycle of struggle/learn/get-new-problem is something that many students experience throughout college as they rarely have time to stop and demonstrate their rapidly growing skillset in a meaningful way.

My motivation for being part of this project is that I want a chance to slow down and use the skills and algorithms I have picked up in my years at UCF in a useful and creative application instead of an abstract, un-impactful thought exercise. Instead of expecting this project to challenge me with highly complicated algorithms and difficult proofs, I expect this project to challenge me in project management skills, team coding, and the application of numerous techniques that I have learned but never had to implement in a meaningful way. There is an important difference between *knowing* and *doing*; and so I am taking this chance to remind myself what all my learning has amounted to.

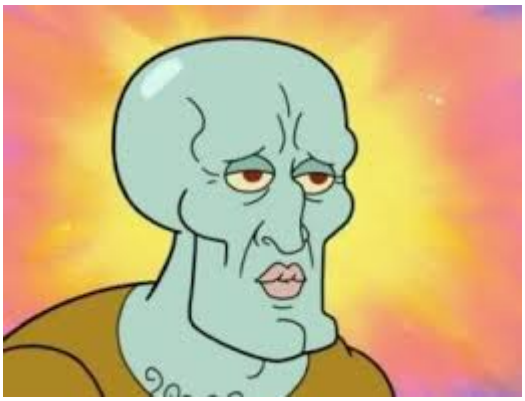
4.3 David Akridge



This project caught my interest due to the combination of the creative nature of Dungeons & Dragons mixed with the real world applications that can be gained by accomplishing the projects end goal. In my time at UCF, we have learned tons of information primarily focusing on theories and concepts of computer science. While plenty of classes have had us working hands on, there is still much to learn before being able to stand out in the real world. This project is going to have us utilize the various algorithm implementations weve learned as well as diving into web development. As we havent covered web development in too far detail in the UCF curriculum, this project easily provides an opportunity for me to grasp the understanding of HTML, Javascript, Angular.js, etc. These are all languages and formats that I have had the desire the learn but lacked the drive to do on my own.

Along with being able to learn practical CS skills, DMBD will allow us to find creative workarounds be it code, web-page design, or even artwork. This is a really important aspect of the project to me, because many of the projects weve had to do prior have been numerically heavy and honestly not all that interesting. An engaging idea could help propel our ability to learn these new concepts, especially if we are interested in them on our own. Now that there is a great project motivating me and a great team to work alongside with, there is no doubt we will learn all the ins and outs of the subject.

4.4 Hunter Figueroa



I have been playing tabletop role-playing games for over 6 years now. They served as a creative outlet where I could experiment with new creative ideas. I prefer acting as a DM, Ive only played a single session as a player, I loved the idea of being all powerful creator of a world that I could share with my friends. The tabletop community, namely the Dungeon & Dragons community is like no other that I have seen before. It is composed of highly talented, highly cooperative and interactive people who do their very best to better and expand the role playing community. For close to two years now I have put a considerable about of my free time into constructing a digital service to help this same cause. So when I heard Jacob pitch this idea, I knew it was the one for me. Weve got something really special here, and with the team that we have I feel we can create something game changing.

User Interface

5.1 Colors

5.2 Icons

5.3 Login

5.4 Register

5.5 Homepage

5.6 Search Page

5.7 Arena Page

5.8 Battlefield

5.9 Encounter Profile

5.10 User Profile Page

Technical Goals

6.1 Overall Goals (Requirements???)

6.2 Project Timelines (Milestones???)

6.3 Communication

6.4 Research

6.5 Budget

Server Design

7.1 User Roles

7.2 Node.js

7.3 Express.js

7.4 Multer

7.5 Mongoose

7.6 Server Sequence Diagrams

Database Design

8.1 Schemas

8.1.1 User Schema

8.1.2 Arena Schema (HAHA THAT RHYMES)

8.1.3 Encounter Schema

8.1.4 Combatant Schema

8.1.5 Obstacle Schema

Application Design

9.1 Design Patterns

9.2 Accessibility

9.3 Types of Views and Controllers

9.4 In-Game Roles

9.5 Map / Environments

9.6 Turn Structure / Order

9.7 Computer Opponent (A.I.)

9.8 Encounter Diagnostics (Stress-Tester)

9.9 Encounter Browser

9.10 Encounter Creator Tools

9.11 Storyboards

9.12 Version Control

9.13 Data Processing from API (WHAT DOES THIS MEAN???)

Research

10.1 MEAN Stack

The creation of developer tools like MongoDB, Express.js, Angular.js, and Node.js (otherwise known as the MEAN stack) forge a collection of technologies that allow for JavaScript functionality. The culmination of these technologies allow developers to create entire web applications using JavaScript.

10.1.1 MongoDB

10.1.2 Express.js

Express.js is the natural choice when implementing Node.js in a project, because Express is the standard server framework for Node. In short, Node and Express were the first that gave developers the tools to build the back-end and front-end software using JavaScript. Express is the framework that actually implements the functions needed to build a website or web application. Server-side code written using Node.js is published and built as a website using Express.js.

10.1.3 Angular 4

10.1.4 Node.js

Node.js is a runtime application that starts small but grows more and more useful as additional modules are implemented. There are a few reasons why it is the first choice when building a web application like this project. Node allows for JavaScript coding on the server side of a website or web application. This results in the idea that code can be written once, but ran on both the server and the client browser, optimizing which functions would give users faster response times while also lifting a bit of work for the server. Writing server-side code in JavaScript using Node also keeps the logic and code very similar across the server and client. Also, Node uses an asynchronous even driven mode, so that when a task is performed, other

tasks are not prevented from being run.

10.2 Bootstrap 4

10.3 RESTful

10.4 Game A.I.

10.4.1 Pathfinding

10.4.2 Basic Rule Following (Non-A.I. Computer Opponent)

10.4.3 Decision Trees

Appendix

Figures

Tables

References