

Zach Motsinger

Philip Read

Aaron Cowdrey

Dungeons and Dragons Management System

INTRODUCTION

For this project, we created a management website for Dungeons and Dragons that would allow the user to track various pieces of data as they played through the game on the tradition pen and paper format. In order to achieve this, we created a server and database that would represent a mini-world that could effectively track all elements of the game. This included Players, Monsters, NPCs, Locations, Items, and Quests, as well as the relationships between these elements. Ultimately, we wanted to help the dungeon master keep track of as much information as possible as he developed the game for the players.

SYSTEM ARCHITECTURE

In the early stages of this project, we intended to use this data to allow the user to play a game using only our website. As our ideas evolved and we began to implement several aspects of our design, it became clear that we were going to be limiting the user in several ways. Namely through the limitation of dialogue options we would be forced to impose to allow the user to interface with a GUI. Dungeons and Dragons normally allows the players complete freedom to do and say whatever they want, and we chose to modify our ideas slightly in order to preserve the true nature of the game.

Because of this, we had to deviate slightly from our initial ER diagram. Originally we planned for players to be able to interact with other players, travel to places, accept or decline quests, store the quests in a journal, pickup or drop items from inventory, and access a detailed record of stats and

abilities. Most of this is still possible, but some of it is now implied through the actual game instead of through our system. For example, characters can still have their inventories updated, and they can still be moved between locations, but things like accepting a proposed quest and interaction between players is left to the dungeon master and the people who are involved in the game.

DATABASE DESIGN

Once we understood this new direction, we built and modified the necessary tables that were derived from our existing ER diagram, and set up a server that was capable of sending queries to pull data from that database. Finally, we designed a website that could handle the transmission of these queries to allow the user to add and update the rows of the already existing tables, and modify the relationship table between various entries.

After researching several outlets, we finally settled on a PHP based website due to its ease of interacting with SQL servers. In order to perform our initial tests, we created a mock SQL request form that allowed the user to retrieve data from the server by typing queries into a text box. This preliminary implementation greatly increased our understanding of what we needed to accomplish in order to move forward.

IMPLEMENTATION

Due to the extensive ER diagram laid out in the early stages of development, the time required to implement our modified database was significantly reduced. Once this was completed, we extended our PHP program to allow the user to interact with the data we had populated in a more logical way. For example, instead of sending raw SQL commands, the user was now creating items and characters through simple web forms.

In our final product, our website first allowed the user to populate the database tables with an unlimited amount of data before starting the game. This data included, Characters, Monsters, NPCs, Locations, Items, and Quests. It was up to the user to decide what elements would be necessary before moving forward and starting the game. Once the game was started, the user was taken to another screen that allowed them to move items between players and the global inventory, move players and monsters between locations, track quests, and update player stats. As the game is played out, the dungeon master will use our system to track the flow of all elements as the game unfolds.

TECHNICAL DETAILS

The website was ultimately divided into two parts; a section for users to create new data and a section for users to enter after they had started the game. We thought this divide was necessary to preserve the game's integrity. For example, items are all given an initial location, and we thought it would be counter-productive for items to be added to a location after the game had begun and players had started to move through the world.

The section that allowed the user to add data to the database was divided into several different pages. On each of these pages, most of the work done using separate forms that encased all of the necessary text fields. Once the submit button was pressed, the data from each of these forms was passed to another page that used it to build standard string-based SQL commands and modify the existing database.

For the Main Menu we wanted to design a way to access data without having to make a separate SQL call each time. In fact the only time we wanted to make an SQL call would be when any data is needing to be updated. To do this we gathered all the relevant data, i.e. character names, locations, etc., into a series of PHP arrays. Then we converted each array to a JSON string and passed them to a series

of echoed Javascript. Then when the website had loaded, the data existed as static JSON objects able to be accessed without query the database each time.

RESPONSIBILITIES

Although all members of the group did a little bit of everything, each member made a larger contribution in certain areas than the others. Zach Motsinger developed the Main Menu pages and the PHP code to update the SQL server. Aaron Cowdrey set up all the SQL tables and their corresponding properties. Philip Read developed the Main View screen and the PHP code to fetch and update any transfers.

TESTING

Through every stage of our project, we were consistently testing everything and constantly trying to break what other members of our group had created. Once the database was live, we would populate it with dummy data and perform standard SQL commands. Once the website was live, we reiterated this action by using the forms and page calls instead of raw interaction with the database. Ultimately we were able achieve a robust system because of the extensive testing put forth by all members of our group.

CONCLUSION

A standard game of Dungeons and Dragons is host to a copious amount of data, and is often difficult for players to track without the use of some kind of medium. It was ultimately our goal to ease the burden of the Dungeon Master, and provide them with a sleek and simple interface that would allow them to focus more on creating and mandating the game in front of them. Through the use of some basic PHP and SQL, we believe that we have achieved a robust and necessary GUI that would allow the user to effectively control a game with our system.

COMMIT LOG

Commits on Dec 11, 2014

cleaning
zmotsing

Commits on Dec 10, 2014

Characters can now be moved between locations. -PR
Philip Read

Added Responsiblites Section to Report. -PR
Philip Read

final report
zmotsing

Added Location Data -PR
Philip Read

fixes to the style to make them work better on the laptop
zmotsing

Fixed Quest text. -PR
Philip Read

Made man changes. Biggest is being able to give items to other player...
Philip Read

Give item shell
zmotsing

Commits on Dec 9, 2014

DROP DOWN MENUS
zmotsing

characters and items must be assigned a location. this uses a dropdow...
zmotsing

Cleaned temps. -PR
Philip Read

Added initial version of mainView. Still needs styling to be complete...
Philip Read

quests have titles
zmotsing

framework for more relationship information
zmotsing

Commits on Dec 8, 2014

added quests, locations, and items

zmotsing

cleaning

zmotsing

cleaning

zmotsing

update to accomodate our new direction

zmotsing

create character now pushes to sql server

zmotsing

Commits on Dec 2, 2014

New Character page - No SQL commands yet

zmotsing

skeleton for new accounts/characters

zmotsing

skeleton for new accounts/characters

zmotsing

cleaning

zmotsing

login page

zmotsing

Commits on Nov 30, 2014

Added login.php

PhilipRead

Added newCharacter.php

PhilipRead

Added mainMenu.php

PhilipRead

Commits on Nov 7, 2014

Project 2 Report Added

Aaron Cowdrey

Initial PHP code for interfacing with database. PR

Philip Read

I DON'T CARE

Cowdrey

Commits on Nov 6, 2014

Added Report 2

zmotsing

Commits on Oct 3, 2014

Delete ERDiagram.pdf

Cowdrey

Added new LR Diagram

Philip Read

Create ERDiagram.pdf

Cowdrey

Commits on Oct 3, 2014

Added new LR Diagram

Philip Read

Delete DND database %283%29.pdf

Cowdrey

Removed Diagram for new one...

Philip Read

First Commit

Philip Read