Dungeon Positioning System

Project Team: Elizabeth Dong, Gregory Purvine, Spencer Whyatt, Phillip Jones, Ray Cockerham

Table of Contents

5. System – Analysis Perspective – *Group responsibility*

- Identify subsystems – analysis point of view
- System (Tables and Description)
    - Data analysis
        - Data dictionary (Table - Name, Data Type, Description)
    - Process models
- Algorithm Analysis
    - Big - O analysis of overall System and Sub-Systems

6. Project Scrum Report - *Group Responsibility*

- Product Backlog (Table / Diagram)
- Sprint Backlog (Table / Diagram)
- Burndown Chart

7. Subsystems

7.1 Subsystem 1 – Name 1 - *Individual responsibility*

- Initial design and model
    - Illustrate with class, use-case, UML, sequence ..... diagrams
    - Design choices
- Data dictionary
- If refined (changed over the course of project)
    - Reason for refinement (Pro versus Con)
    - Changes from initial model
    - Refined model analysis
    - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint - Link to Section 6)
- Coding
    - Approach (Functional, OOP)
    - Language
- User training
    - Training / User manual (needed for final report)
- Testing

7.2 Subsystem 2 – Name 2 - *Individual responsibility*

- Initial design and model

    - Illustrate with class, use-case, UML, sequence ..... diagrams
    - Design choices
- Data dictionary
- If refined (changed over the course of project)
    - Reason for refinement (Pro versus Con)
    - Changes from initial model
    - Refined model analysis
    - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint - Link to Section 6)
- Coding
    - Approach (Functional, OOP)
    - Language
- User training
    - Training / User manual (needed for final report)
- Testing

7.3 Subsystem 3 – Name 3 - *Individual responsibility*

- Initial design and model
    - Illustrate with class, use-case, UML, sequence ..... diagrams
    - Design choices
- Data dictionary
- If refined (changed over the course of project)
    - Reason for refinement (Pro versus Con)
    - Changes from initial model
    - Refined model analysis
    - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint - Link to Section 6)
- Coding
    - Approach (Functional, OOP)
    - Language
- User training
    - Training / User manual (needed for final report)
- Testing

7.4 Subsystem 4 – Name 4 - *Individual responsibility*

- Initial design and model
    - Illustrate with class, use-case, UML, sequence ..... diagrams
    - Design choices

- Data dictionary
- If refined (changed over the course of project)
    - Reason for refinement (Pro versus Con)
    - Changes from initial model
    - Refined model analysis
    - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint - Link to Section 6)
- Coding
    - Approach (Functional, OOP)
    - Language
- User training
    - Training / User manual (needed for final report)
- Testing

8. Complete System – *Group responsibility*

- Final software/hardware product
- Source code and user manual – screenshots as needed - Technical report
    - Github Link
- Evaluation by client and instructor
- Team Member Descriptions

This is just a guide, and use it to create/improve your report. Feel free to add sections. You are responsible for your own subsystem/s, not other members. You have to contribute to the team's goals and objectives, and develop your subsystem/s, write your documents and slides.

Project Definition

Playing tabletop RPGs using only pencil and paper can be very time consuming. Keeping track of all the tiny details wastes a lot of time that could be better used playing the game. Our system allows for all of this boring bookkeeping to be handled automatically. It will also allow for players to play tabletop RPGs across long distances, so that friends don't need to travel to meet in person in order to play their favorite tabletop RPG.

We want to create this software so that people who want to play games can spend more time doing just that, and less time doing behind the scenes work. Having that behind the scenes work done by a machine will allow the players to spend more time fully enjoying the game. We would also like to incorporate an online aspect, so that friends far away can play together without having to meet in person.

We will build a system that keeps track of players' character sheets and their locations on the game board. It will include panels that display a person's character sheet and display the game board, which will include the locations of all player-characters and non-player-characters currently in play. The system will be able to handle character movement along the game board. It will also give the Dungeon Master to make changes to the game board on the fly. This will allow them to add things to the game board during the session, like enemies or obstacles.

Project Requirements

Functional Requirements

1) Maps: The system will display a map to all players and the Dungeon Master, so that everyone can see where all players, enemies/non-player-characters, and terrain are on the field. The system will have the ability to randomly generate a map for the session, or allow the Dungeon Master to create/import their own map. The Dungeon Master will have the ability to make changes to the map during the game session.

2) Characters: The system will display a player's character sheet. The Dungeon Master will also be able to see the character sheet for any non-player-characters or enemies on the board. The system will be able to save a person's character sheet, so that it may be used again. It will also allow for users to create new characters. They will be able to either create a new character using preset stats for their particular class, be able to manually input stats for their character, or roll for their stats using the built in dice roller.

3) Game Sessions: The system will keep track of individual game sessions. A map, along with all tokens on the map, will be saved as part of that session. The Dungeon Master will have the ability to create new sessions. Players will then be able to join the session their Dungeon Master has created.

4) Dice Roller: The system will have a built in dice roller. The user will be able to choose between 7 types of dice to roll, and the system will use a random number generator to simulate the rolling of these dice. The user will have the ability to roll 4, 6, 8, 10, 12, 20, or 100-sided dice.

5) Game Log: The game log will display the results of all rolls using the built in dice roller. It will also describe all movement on the game board and all additions to the game board made by the Dungeon Master. It will also function as a text chat for the game, allowing players to communicate with each other. This text chat should also add to the role-playing aspect of the game, as characters will be able to talk to one another.

Usability

The system will have a graphical user interface, to allow for ease of use for the user. The GUI will include sections for the game board, players' character sheets, the built in dice roller, and the game log/text chat. In terms of performance, the system needs to be efficient enough so that users do not experience any extraordinary delays or lag in gameplay. Delays are theoretically acceptable (though not desired), so long as they do not interfere with the users' game experience.

System

Hardware: Computer with Java installed, 2GB RAM
Software: Java
Database: MySQL

Security

Security is not a major concern for our application. Our main method of security is ensuring that only the players and Dungeon Master are allowed to join the game session. To that end, we believe that having a randomly generated code be necessary for joining a game session may be sufficient. The code would be randomly generated by the Dungeon Master upon creation of the session. The Dungeon Master would then be able to share it with their players, who would use that code to join the session.

Project Specification

Domain

Hobby, tabletop, entertainment

Libraries / Frameworks / Development Environment

Other than the java.net package of J2SE APIs, we have not yet found any libraries we plan on using, but we are actively searching for libraries that will make implementing our system simpler. We plan on developing our solution with a Java backend, along with a MySQL database. We are still looking for the ideal combination of tools to create our system.

Platform

Our system is built for use on a desktop platform. We believe this platform best suits the needs of our system. A mobile application would not be able to contain the information as well as a desktop, due to the smaller screen size. We also believe that making the system a desktop application will allow us to reach a wider audience, given our time frame. To properly reach a mobile audience would likely necessitate making iOS and Android applications, while a desktop audience only requires making one web application that can be run on all environments.

Genre

Application