Dungeons and Dragons Displayer

By: Parker B, Sean G, Jamie P, Jeffrey R, Quinten S. CPRE 186 Group D2

Our Goals

- Create a digital board for the popular RPG Dungeons and Dragons using a computer monitor and a Raspberry Pi, with the ability to import characters, monsters, and maps into the board, and run the scenarios

- Make the “game” be capable of running on an OS as simple as Raspberry Pi, and have the ability to share premade scenarios and characters so multiple users can have exciting, unique experiences.

General Design

- We have 3 main classes which control our program.
- Each new game creates an instance of a StartBoard object, which also creates an EditGame object, which sets the game to a default scenario.
    - The game editor allows the user to change the map and place monsters on the board.
- Once the user deems the game ready to play, the Play button starts the StartBoard class, which creates a game using the updated version of the EditGame object (or uses the default map and monster setup if the user chooses not to change anything).
- Originally, we used the Java Lightweight Game Library, though this was found to be incompatible with other features we wanted to use for different parts of the game. Our project runs on 2DGraphics and JFrame.

In-Depth Design

- When the player selects “Start Game,” a new instance of JFEditGame is created to set up a basic version of the DnD board, which includes a default map layout and a default player icon.
    - Choosing “Select new map” brings up a new screen where the player can select an image to display as the map and also character selection.
        - This new map selection overrides the default one, and can be re-chosen as many times as the user wants.
    - Choosing “select new character” brings up a similar window to the map selection, but this one changes the player icon, and will override the default player icon if a new character is selected.
    - Starting the game prints the map with fog of war, and places the character in the top left corner of the screen. Wherever the player is on the map, the fog of war will disappear, leaving a trail on the map where the player has discovered
        - The fog of war was made using a 2d graphic drawn on top of the map image, with the player model having a small radius that “erases” the fog of war image, revealing the map beneath.
- Selecting “Exit” will quit the game, believe it or not.

Resources We Used

- Old Dell desktop, turned raspberry pi
- Someone’s monitor (either ones available in the labs or a borrowed one)
- Github, Git Bash
- Oracle.com (java class and method implementation)
- StackOverflow
- Eclipse
- Previous knowledge of Java

Hardships

- Booting the desktop in the Raspbian Operating System took considerably more time than expected
- When one or two people were making large, sweeping changes to the code, others couldn’t make smaller changes necessary until the big ones were finished (and pushed to github), slowing down our progress.
- Lack of direction; we started in a good place but once we hit a difficult task or had to implement a new feature that we could not grasp code-wise, our work slowed significantly.
- To get the features we wanted, we had to look into previously unknown areas and implementations of java.

What We Learned

- To scale back our ideas. Our project is significantly smaller than what we hoped for, and has fewer features.
- How to use github, and why it is crucial to making projects in large groups work.
- Git bash can be frustrating to use when not understood.
- The importance of communication among group mates to accomplish a common goal.
- Writing code using methods and classes we had no previous experience with led to a longer development time, and less focus on the hardware side of the project.

Final Thoughts

- Larger projects, even something as simple as a DnD board, take much longer than expected, and to get all the features we wanted, it would have taken more focus and drive on our individual parts to fully implement them.
- When in doubt, the best resource to pull from is the community and its ideas. Brainstorming with group members led to new ways of approaching our problems, some of which were good enough to be used in our code.