# Simple Console Game Design Document

## **Part 1: The Basics**

This stage consists of a very basic

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Details** | **Completed** |
| 1.1 | Simple Menu | A menu that appears at the start of the program. It provides option to Start the Game or Quit the Game. | YES |
| 1.2 | Game Loop | A simple game loop in which the Hero and the Boss attack each other until one is defeated. An introduction happens before and a conclusion happens after. | YES |
| 1.3 | Agent Models | There are the models for the main ‘participants’ of the game: Hero (controlled by the player) and Boss. Each of them should have the following:   * Hit Points * Mana Points (unused for now) * Name * Description * A set of attacks   When an Agent’s HP runs out, they die and the game provides the consequences of this. | YES |
| 1.4 | Endings | There will be two endings from the game, one when the Hero wins and one when the Boss wins. | YES |
| 1.5 | Battle System | The battle system will consist of attacking and defending. Attacking does damage whereas defending will reduce damage. The Player will have one choice of attack whereas the Boss will have two attacks that they will choose randomly. | YES |
| 1.6 | UI | The UI will consist of a text-only interface with no sounds or music. During the game loop, the user will be given a simple menu interface to choose their commands. After choosing their commands, the consequences will appear as lines of text on the screen. The stats of the Hero will appear, while the stats of the Boss will remain hidden. | YES |
|  |  |  |  |

## Part 2: Further Refinement of Basics

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Details** | **Completed** |
| 2.1 | Game Loop | Further addition to the game loop: Indicators for the Hero and Boss when they are at high, medium and low Hit Points respectively. | YES |
| 2.2 | Text System | A system that stores the strings used in the game as raw text and categorizes them based on their placement in the game (move text, intro text, combat text etc.) | YES |
| 2.3 | Text Load System | A system that loads the text above into the correct Objects and Classes. One option to consider is JSON or XML. | YES |
| 2.4 | Basic BOSS AI | The Boss now randomly chooses between a selection of moves instead of just one move. No A\* pathing or such for now. | YES |
| 2.5 | Switch to MVC system | The game’s internals should follow the Model View Controller system. The user inputs content to the view. This gets passed along to the controller, who then acts to update the models, then returns the appropriate information back to the view. | YES |
| 2.6 | Menu for Attacks | The User should be given a menu in the game showing what options are available. They can enter numbers or the word itself to perform the move. | YES |
| 2.7 | Additional Attack | The Hero should be given an additional attack that costs MP but does more damage as a result, to fit the above. | YES |
| 2.8 | Run Option | The Hero should be given a flee option as well, which has a random chance of working. | YES |
| 2.9 | Move Model Overhaul | The Move model should be reworked to fit non-lethal moves as well (such as the Run option). | YES |

## Part 3:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Details** | **Completed** |
| 3.1 | Debug Mode | Add a Debug Mode to the game. This will show the stats of both Agents during the game loop. It will help with beta testing. | YES |
| 3.2 | Improve Interface | The current interface is clunky and difficult to look at from a console perspective. Make it less cluttered. |  |
| 3.3 | Refactor View to Controller code | The current communication to View to Controller is based on Boolean values. However, game feedback may not always fall between two choices. An enum system to categorize these game feedback types should be created. |  |
| 3.4 | Moves stored in JSON | To improve how moves are created and inserted, they should be stored inside a JSON file. |  |
| 3.5 | Game text stored in JSON | To improve how game text is created/inserted, it should be inserted inside a JSON file. |  |

## Part 4:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Details** | **Completed** |
| 4.1 | Poison Attack | Implement the POISON status in game. Poison should decrease a set amount of damage (based on percentage) to an Actor per turn. This may require implementing a function that checks for these ‘idle’ state changes. |  |
| 4.2. | Implement Talk action | The Talk action is meant to be a non-damaging way of communicating with an opposing Agent. A list of options should appear to the user and depending on what they choose different actions should be taken.   * Boss gets stronger * Boss gets weaker * Secret ending |  |
| 4.3 | Implement sound. | Add a basic sound attack whenever a move is conducted.   * Attacking sound for Attack * Magic sound for Magic Attack * Menacing sound for boss Attack   Add move |  |
| 4.4 | Implement basic AI pathing | The Boss should perform a different move based on the POISON status and whether or the choices from the Talk action. |  |
| 4.5 | About page | Put a basic about page in the menu. Edit the menu to make it not rely on hardcoded numbers. | DONE |
|  |  |  |  |
|  |  |  |  |