Olde Worlde Phunne – Maze Game

Design Specification 29/07/2019

Version 1.0

Contents

[1. Introduction 3](#_Toc15416917)

[1. Requirements 4](#_Toc15416918)

[1.1. Core Functionality: 4](#_Toc15416919)

[1.2. Maze Structure: 5](#_Toc15416920)

[1.3. Game Items 6](#_Toc15416921)

[1.4. Overall Program Flow: 7](#_Toc15416922)

[1.5. In-Game Program Flow: 8](#_Toc15416923)

[2. Mock-Ups 9](#_Toc15416924)

[2.1. Enter name 9](#_Toc15416925)

[2.2. Read configuration files 10](#_Toc15416926)

[2.3. In-Game Loop 11](#_Toc15416927)

[2.4. Game Finished 12](#_Toc15416928)

[3. Data Formats 13](#_Toc15416929)

# Introduction

Olde Worle Phunne games has requested a new video game for it’s website to attract more visitors. As per the provided design documentation, this game must be a simple maze-based adventure game in which a single player character is able to progress through a series of rooms, collecting money and avoiding obstacles on the way in an attempt to reach an exit point with the most money possible.

As the purpose of this game is to increase traffic to Olde Worlde Phunne’s website, and considering the week long development timeframe, the game shall be simple, accessible and easy to play for users. With these points in mind, the following requirements have been derived from the Project Definition document.

As no format, visual or interaction style has been specified for the game, it is assumed in this document that the initial version of the game shall be a command-line base text adventure, in which the user inputs commands and the game responds with text explaining the updated state of the game.

# Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Requirement** | **Further Details** | **Comments/Questions** | **Phase:** |
| Core Functionality: | | | | | |
|  | The Maze Game shall present a user with a maze-based adventure challenge, in which a maze of rooms is generated for the player, the player can make on how to progress through these rooms and also perform actions within then and ultimately must reach an end goal. |  |  | **1** |
|  | The Maze Game shall have a completion state, following the user having reached the exit point of the maze. |  |  | **1** |
|  | The Maze Game shall have a fail state, in which the user has died before reaching the end of the maze. |  | MS – this is not outlined in the requirements documentation, but is instead my idea for a potential future project improvement to make the game more interesting. | **2** |
|  | The Maze Game shall allow users to reset their Maze Instance at any point during play. |  |  | **1** |
|  | The Maze Game shall generate a Maze based on configuration values read from a text-based configuration file. |  | MS – The design documentation specifies that the data format for this configuration file should be a text file. I intend to use a .json file, as Json is a data format I am familiar with and one more suitable to a c#, .net based application due to its structure and .net library support. In a real-world scenario, I would discuss this decision with the project documentation author. | **1** |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Maze Structure: | | | | | |
|  | A Maze shall consist of a number of rooms. |  |  | **1** |
|  | A Room shall a number of exit passages (between 2 and 4), with all exit passages connecting to another room, except for the final exit passage which shall exit the Maze. | Exit Passage directions are:   * North * East * South * West |  | **1** |
|  | A Room shall contain a number of interactable Items. | Items consist of collectables, such as money, and enemies. |  | **1** |
|  | One room within the Maze shall contain the final exit point of the Maze, which is the end-goal of the game. |  |  | **1** |
|  |  |  |  |  |
|  |  |  |  |  |

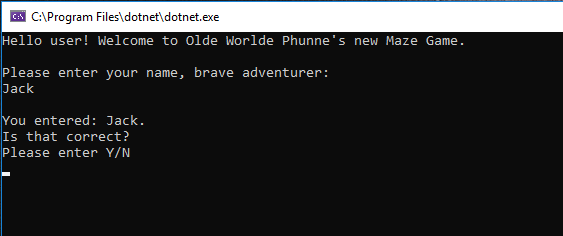
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Game Items | | | | |
|  | An Item shall be one of two types – Threat or Treasure. |  |  | **1** |
|  | A Treasure item shall be collectable by the player. |  |  | **1** |
|  | A Threat item shall be destroyable by the player. |  |  | **1** |
|  | The player’s total amount of money shall be recorded in the game |  |  | **1** |
|  | Enemies shall be able to steal money from the player. On stealing money, the player’s total amount of money shall be reduced. |  |  | **1** |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Overall Program Flow: | | | | |
|  | The user shall be able to start the game executable file. |  |  | **1** |
|  | The User shall be able to enter their chosen player name. |  |  | **2** |
|  | The program shall read the contents of the configuration file. |  |  | **1** |
|  | The program shall alert the user of any errors encountered whilst reading and parsing the configuration file. |  |  | **1** |
|  | The program shall generate a Maze based on the values in the configuration file. |  | MS – I intend to use some elements of randomisation for the Maze generation algorithm. This should be done using a seed, so that the user can recreate their previous mazes should they wish to attempt them again. | **1** |
|  | Once a new Maze has been generated, the use shall be able to start a new instance of a game. |  |  | **1** |
|  | The user shall be able to restart their game-instance at any point during the course of the In-Game loop. |  |  | **1** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

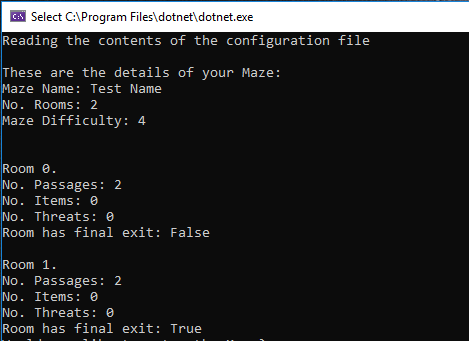
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| In-Game Program Flow: | | | | | |
|  | On beginning the in-game loop, the game shall present the user with introductory text. | The introductory text shall be a short description of the scenario the player is in, and their goal. |  | **1** |
|  | The user shall begin the game from any one of a random selection of Maze Rooms. | The potential Maze Rooms that the user can initially start the game from shall not include the final room that contains the exit passage. | MS – I don’t think it is a good design possibility to allow the user to begin the game from the room with the exit passage. Added comment in addition details to explain that this shall not be the case in the design. | **1** |
|  | The game shall present the user with a list of available user-input commands at all points the user is able to perform a command. |  | MS – Providing the commands every time the user is able to perform an action will prevent confusion about what the user is able to do. | **2** |
|  | If the user enters a command that is not recognised, the game shall inform the user that their command is not recognised, and present the user with the command list. |  |  | **1** |
|  | On user entry of a successful command, the game shall perform the appropriate action and provide the user with feedback on how their scenario has changed. |  |  | **1** |
|  | On entering a Maze Room, the game shall present the user with a description of the room they are currently in. | The description of the current room may include the following hints:   * Whether the room they have entered is closer to the exit point of the Maze or not. * What items may be found in this room |  | **1** |
|  | The user shall be able to progress to different rooms through the use of passages. |  |  | **1** |
|  | If the user does not clear all enemies from a room before attempting to leave the room, they shall lost treasure. |  | MS – For future developments, we could look at adding a health systems, separate from the Treasure collection system. | **1** |
|  | If the user reaches the exit point of the maze, the game shall present a summary of their progress | This summary shall show the number of rooms traversed, as well as the amount of money the player has accumulated. |  | **1** |

# Mock-Ups

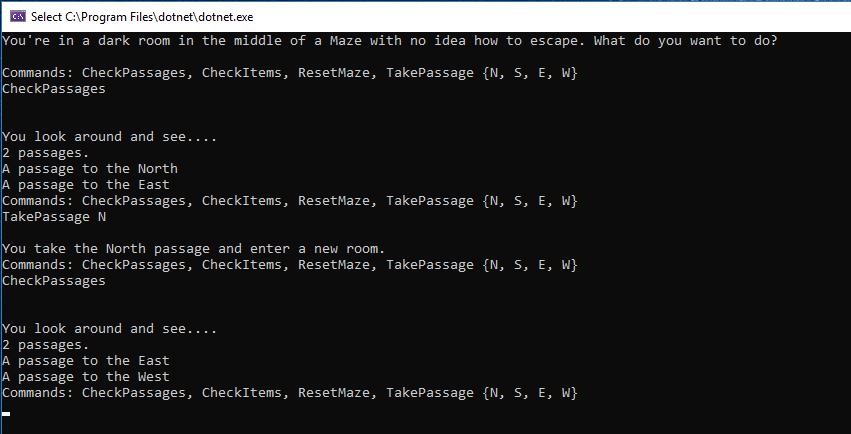
## Enter name



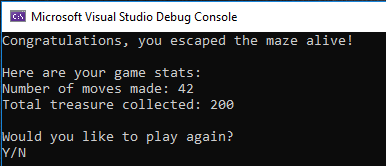
## Read configuration files



## In-Game Loop



## Game Finished



# Data Formats

Json for the