Game Design Document:

Title: Assignment #4: Alice's Adventures in Wonderland

Halie Favron

Student ID: 3673820

COMP 306

November 29, 2024

Game Concept

Title: Alice's Adventures in Wonderland

Genre: Text based adventure game

Summary:

A text-based adventure game, based on "Alices Adventures in Wonderland", by Lewis Carrol. The user plays as Alice. The player tries to make friends, collect treasures and solve puzzles as they navigate through a series of locations from the book. The locations are often filled with characters and items (gifts, treasures, key items, weapons, and keys).

Gameplay Overview

Player Goal 1: Make friends with the characters of Wonderland, ie. The Queen of Hearts

Player Goal 2: Collect treasures, ie. The Hookah

Gameplay Loop:

The player enters a location. They then can interact with the items and characters inside the location by typing in their user input. Until they leave the location, then the next location is loaded and the loop repeats. This goes on until the player loses/ wins/ or ends the game

Game Design Elements

Story and Setting:

- The narrative begins with Alice waking up in a meadow with subtle hints that things are “not quite right.”
- The game begins when Alice follows the White Rabbit and falls down a rabbit hole. Once she does, she can no longer return to the meadow but she is in a network of locations such as the Rabbit Hole, Long Hallway, Cottage (a safe room), Hare’s Lawn, Mushroom Forest, Pool of Tears, Isle of Sorrow, Garden, and the Castle.
- Locations usually contain items and characters to interact with. Most of the characters dialogue is inspired/paraphrased or directly quoted from Alices Adventures in Wonderland.

Player Role and Objectives:

- Exploration: The player navigates between different locations, with each area having its own description, characters, and interactive items.
- Interaction: Interacting with the characters and items of Wonderland is central to the game play.
- Inventory & Treasure Management: The player collects items such as keys, gifts, treasures, and other key objects that help unlock new areas or boost her abilities. Certain treasures can be stored in the Cottage for additional points. While others can be given to characters as gifts to boost the players overall friendship score.
- Goal: The player’s ultimate objective is to explore the fantasy world, befriend (or at times confront) the characters. The player accumulates a high overall score based on friendship with the characters and collecting/storing treasures in the cottage safe room.

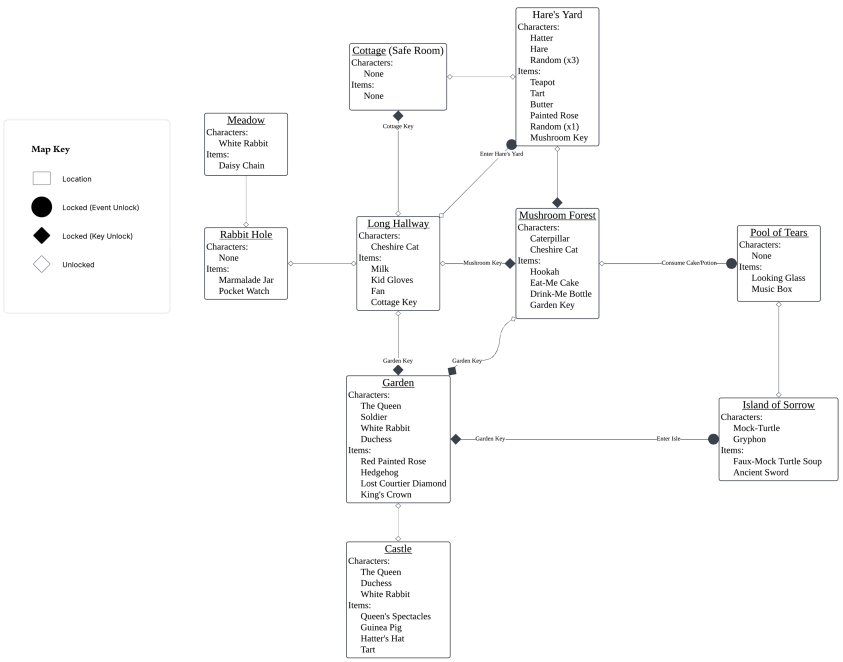
Game Objects:

Locations:

- Meadow
- Rabbit Hole
- Long Hallway
- Cottage (Not from book)
- Mushroom Forest
- Hares Lawn
- Pool Of Tears
- Isle Of Sorrow (Not from book)
- Garden
- Castle

Characters:

- Cheshire Cat
- The Queen of Hearts
- Mad Hatter
- March Hare
- White Rabbit
- Caterpillar
- Duchess
- Soldier
- Gryphon
- Mock Turtle



Items

From Book:

- Milk
- Butter
- Faux Soup
- Pocket Watch
- Kid Gloves
- Painted Rose
- Tart
- The Queens Spectacles
- Daisy Chain
- Looking Glass
- Music Box
- Fan
- Hookah
- Hedgehog
- Guinea Pig
- Teapot
- Courtier Diamond
- Kings Crown
- Hatters Hat
- Drink Me Bottle
- Eat Me Cake
- Marmalade Jar
- Garden Key

Items:

Not From Book:

Cottage Key

Mushroom Key

Ancient Sword

Core Gameplay Mechanics

Exploration and Movement

Locations:

- The game is divided into distinct locations (e.g., Meadow, Rabbit Hole, Long Hallway, Cottage, etc.). Each location has a unique set of descriptions, available exits, items, and characters.

Exits and Directions:

- Players move from one location to another by entering commands such as "enter north" or "go east." Some locations have exits that are locked, requiring a specific key or an event to occur to unlock them.

Exploration Command:

- The player can use commands like "explore" to get a detailed description of the current location—including visible characters, items, and exits.

Interaction and Command Parsing

Command Parsing:

- The game parses player input into structured commands. It categorizes words as keywords, verbs, items, characters, or directions.
- For example, commands may include "pick up marmalade jar," "talk to Cheshire Cat," "fight Duchess," or "examine music box."

Action Checks:

- Each command is verified against a series of checks (e.g., valid inventory, proper direction, appropriate item usage). Customized error messages or suggestions are provided if the input is invalid.

Core Gameplay Mechanics (Continued)

Inventory and Item Management

Player Inventory:

- The player has an inventory that holds items picked up during the adventure. Items may affect Alices stats, such as attack damage, or be used to interact with the world (for example, keys unlock exits). Most items are either gifts or treasure.

Treasure Inventory (Cottage):

- When Alice reaches her safe room (the Cottage), she can deposit collected treasures. This mechanism boosts her overall score and is one of the main goals of the game.

Item Types:

- Items are categorized by type (e.g., Key, Gift, Treasure, KeyItem, Weapon). Certain items (such as keys) are triggered by events rather than manually picked up by the player.

Character Interaction and Combat

Dialogue and Friendship:

- The game includes multiple characters (e.g., White Rabbit, Cheshire Cat, Caterpillar, March Hare, Mad Hatter, Queen, Duchess, Mock Turtle, and more).
- Gift choices may lead to improved relationships (friendship) or the risk of turning an ally into an adversary.

Combat:

- In some situations, if Alice interacts incorrectly or antagonizes a character, combat ensues.
- Both Alice and the adversary exchange attacks until one's hit points reach zero.
- Victory results in the enemy fleeing and removal from the current location; defeat ends the game.

Technical Design

Object-Oriented Design:

The game uses an object-oriented design with modular classes. This design promotes organization, maintainability, reusability, and clear interactions among components.

Classes

Game Class

Acts as the central class. It uses the other classes (Control, Locations, Characters, Items, Inventory, and Alice), loads game objects from files. It manages the entire gameplay loop, loads locations (sets the current location) and their corresponding items/ characters, it executes the location functions, checks input, and executes the action the player wants to perform etc.

Control Class

It gets, validates and parses player input. When it parses player input it identifies and categorizes the tokens (individual words). Then it returns an integer array populated with representations of the users choices. It is used by the Game class to process player actions.

Actions Class

The action class stores all possible actions in maps. It is responsible for parsing the Actions.txt file and storing the actions in the appropriate enum/ string maps. These maps are then used by the Control class to sort player input

GameObject Class

Serves as the base class for Items, Characters, and Locations. It is responsible for managing common attributes (name, short/ long description), as well as

Classes (Continued)

Locations Class (Derived from GameObject)

The Location class manages the games environment, it manages the locations possible exits, a hint about the location, and whether or not the room has been entered.

Exit Class

The Exit class manages data for the possible exits of a location, it manages the name of the next location, a description, keys (if applicable), the direction which the exit is located, and whether or not the exit is unlocked. It is used by the Location class. Which stores a vector of possible exits for. each location.

Characters Class (Derived from GameObject)

The Character class represents non-player characters (NPCs) from the book. It manages their friendship score, their hit point, their attack damage and which gifts they like/ and don't like.

Items Class (Derived from GameObject)

The Items class represents game items it manages the item type, if it can respawn (always either true or false), respawn (always starts true, then changes depending on whether or not an item is can respawn and if it has already been picked up), points, etc

Inventory Class

The Inventory class is used to manage a vector of items (players/treasure inventory), including adding, removing, searching, displaying, and validating.

Alice Class

Represents the player character. It manages the different scores (the total score, treasure score and friendship score), hit points (hp), and attack damage.

Text Files

Locations File

The Locations.txt file contains the data needed for each location. Each location data block contains the location's name, short description, long description, any exits the location has, and a hint for the player about the location

Items File

The Items.txt file contains the data needed for each item. Each item data block contains the item's name, short description, long description, item type, and if applicable points for treasure.

Characters File

The Characters.txt file contains the data needed for each character. Each character data block contains the character's name, short description, long description, item type, and if applicable points for treasure.

Action File

The Actions.txt file contains all possible words that can be input by the player in order to control the game. It is broken up into data blocks, ie. Keywords, Verbs, Characters, Items, Locations, and Directions. Then the label of each set of words maps to an enum, and the rest of the line is made up of possible words that can be input separated by spaces (this comes with the limitation that the player must separate words with dashes)

Plan

The game begins with Alice (the player) falling down the rabbit hole, entering the world of Wonderland. From there, the player explores different locations, encounters characters/items, and solves puzzles to unlock new locations.

Each location will enable the player to interact with characters and objects that either increase the score, world build, or unlock new areas.

The story will be inspired by Alice in Wonderland by Lewis Carroll with key locations, characters and items from the book.

The game ends when the player wins/ loses or ends the game.