CSC111 Winter 2024 Project 1

Naoroj Farhan, Fiona Verzivolli

February 10, 2024

Enhancements

1. Enhancement 1: Items with inheritance and polymorphism for puzzles

    - Our item enhancement uses the ItemFactory class with inheritance to add more functionalities to specific items. We have a main items class which was part of the starter code and a separate class for each item we intend to use in the game, such as the Flowers and IceCreamSandwich class that inherit from the main Items class. We used inheritance here as part of a larger feature that allows the player to interact with items to solve puzzles by typing 'deposit [item name]' or 'use [item name]'. The item class has an abstract method called `print_statement` which for each item, prints a custom message to the player using polymorphism. For items like Key, Tcard, or LuckyPen, `print_statement` simply tells the player whether or not the item has been used successfully. However, for items like Flowers or IceCreamSandwich, when used correctly, the player is rewarded with a hint like 'answer 42 to Ilias puzzle' and can advance successfully in the game.
    - Complexity level: Medium
    - Reasons you believe this is the complexity level (e.g. mention implementation details, how much code did you have to add/change from the baseline, what challenges did you face, etc.): We believe that the complexity level is difficult as we had to write many lines of code and alter the way our items initially worked for this enhancement. This was not included in the baseline/start files, and we faced challenges implementing it, having to alter the rest of our code and incorporate inheritance into our project.

2. Enhancement 2: Puzzles

    - The way that our game is structured involves many small puzzles that include having to unlock a room which contains your tcard, search a garbage can for a key that may open that room, search a rocket in Myhal to find your pen, and do small side-quests to get hints to find items like cheatsheet or pen such as giving flowers to Tom and ice cream to a homeless man. To add to the difficulty of the game, our player has a limited inventory, which means that they need to be strategic of what they're picking up/dropping. These puzzles make the game more exciting and strategic.
    - Complexity level : Difficult
    - Reasons you believe this is the complexity level (e.g. mention implementation details): There were several puzzles/objectives/quests we had to implement, each taking a reasonable amount of time. We also had to be mindful of how each side quest/puzzle related to each other and the order the player must go through them, which added an element of challenge. Testing our puzzle for errors was also a hassle, as each part was connected to one another. Moreover, puzzles like searching a key to open a door or searching a location involved adding data to our locations for whether or not a room was locked or still searchable.

3. Enhancement 3: Extra Functions/Methods

    - In order to progress through our game, we made the following functions/methods: "search," "unlock" and "use" These methods were implemented as a way to interact with items that are required for our puzzles. For example, "give ice cream" when said in a specific room rewards the player with a hint on where to find their pen but when used elsewhere, it gives admonishment such as 'cannot use here!'.
    - Complexity level : Medium
    - Reasons you believe this is the complexity level (e.g. mention implementation details): The complexity level is medium as we had to implement several methods spanning the world, player, and item class files, each requiring careful consideration of game mechanics and our user's interactions. We needed to make sure that each method is compatible with each other and other methods, and had to go through error handling to prevent unexpected behavior. While not too hard, implementing these extra methods took consideration and lots of planning.

4. Enhancement 4: Text Parsing

    - Because player inputs can have typos or similar to the desired input but not quite (think about words like "move" and "go" or 'give' and 'use'), we included a text parsing function to deal with this technicality. We believed that this added versatility was essential to a good text adventure game so we put a lot of consideration into synonyms and similar words
    - Complexity level : Easy
    - Reasons you believe this is the complexity level (e.g. mention implementation details): The complexity level is easy, as we only need to implement one function that then deals with the player's input. There was a tedious part which was generating a word bank, but otherwise, it was not hard to implement.

Extra Gameplay Files

If you have any extra `gameplay#.txt` files, describe them below.