

CSC111 Winter 2024 Project 1

Akram Klai & Reena Obmina

February 4, 2024

Enhancements

1. Information / Clue (The Puzzle)

- The information regarding the location of missing items and access codes (to lockers at Goldring, locks in a stash, and a phone number) can be found after talking to NPCs. These clues add complexity to the game, requiring players to solve math problems to decipher them. Players then use this code to unlock the required items (T-card, Cheat Sheet, and Lucky Pen).
- Medium
- This puzzle introduces added interactive dialogue with NPCs, asking players to solve math problems to decipher these clues. Integrating this feature meant extending the game logic to include puzzle elements—specifically, generating and validating the solutions to math problems that lead to unlocking essential items like the T-card, Cheat Sheet, and Lucky Pen. Implementing this puzzle mechanism involves creating a new layer of interaction within the NPC dialogue system, as well as modifying the Item class to accommodate the unlocking mechanism based on codes derived from puzzle solutions.

2. NPCs

- NPCs provide clues to aid the player in completing the adventure game, including the location of missing items and access codes.
- Low
- This addition of NPCs is considered low complexity, especially because the structure for managing NPC dialogues in our game closely mirrors that of the existing location data management, as outlined in the npc.txt and location.txt files. This similarity allowed us to leverage the already established code framework for locations to easily incorporate NPC dialogue, requiring only minor modifications. By reusing the parsing and data handling logic from the method load_locations, we could efficiently implement NPC interactions within the same system. Essentially, this meant that the groundwork for loading, storing, and accessing data was already laid out, allowing us to focus on the content of the clues rather than the mechanics of data management.

3. Inventory Management

- Introduces a maximum weight limit to the player's inventory, allowing them to carry only two items at a time. If the player exceeds the weight limit, they must drop an item to pick up a new one.
- Hard

- This feature is of medium complexity because it significantly alters the game's inventory mechanics, requiring both conceptual and technical adjustments to the player's interactions with items. To implement this feature, the existing Player class must be modified to include logic that tracks the number of items carried and enforces the weight limit. Additionally, this enhancement necessitates changes to the methods that manage item pickup and drop actions, ensuring that the game prompts the player to make decisions about which items to keep or discard when the limit is reached.

4. Nitpicking

- Access to certain locations is restricted until specific conditions are met, such as collecting essential items. For example, the Exam Centre is inaccessible until all three items are collected, adding strategic depth to the game. Places like Robarts Library and Goldring Athletic Centre have special actions ([look]/[speak]/[collect]) exclusive to those with a T-card in their inventory.
- Hard
- This enhancement is highly complex, primarily due to the incorporation of conditional logic and sophisticated use of inheritance. To facilitate this, the game architecture is enhanced by the addition of a RestrictedLocation subclass, which inherits from the Location class, specifically designed to handle entry prerequisites. This refinement meant modifications to existing class methods that (get_description, update_available_actions, get_available_actions, and update_access) govern player movement and interaction with locations, to seamlessly integrate item checks into the player's inventory. Such an approach enriches the gameplay with dynamically adjusted game mechanics based on player progress.

5. Name Manipulation

- At the beginning of the game, players are given the opportunity to input their names, enabling the Non-Playable Characters (NPCs) dialogue to dynamically adjust and address them on a personal basis. This feature significantly enhances the immersion and fosters a deeper, more personal connection between the game and the player.
- Low
- This enhancement involved adding a simple input prompt for the player's name at the beginning of the game within the main block of our code. Adjusting NPC dialogues to include the player's name is straightforward, requiring zero changes to existing code, specifically within the talk_to_npc method of the Location class to interpolate the player's name into the dialogue strings. No significant challenges were expected during implementation, as it involves basic input/output operations and string manipulation, which are well-supported operations in Python.

6. Information Menu

- At the outset of the game, players are shown an introduction to the game's storyline and objectives, setting the stage for their thrilling adventure. Additionally, with each action the player undertakes, they are seamlessly guided by a helpful prompt: "(type in 'menu' for a list of available commands and 'help [command]' to find how to use a command)". This ensures that players feel supported, especially in moments of uncertainty when they seek clarification on the functionality of specific commands before proceeding.
- Low
- Much like the previous enhancement. This enhancement involves displaying static information at the beginning of the game and providing prompts during gameplay. It required minimal changes to the code in adventure.py and did not alter existing mechanics.

Puzzle Mechanism Involving Inheritance

Add at least one puzzle that uses inheritance in some, reasonable way OR at least one puzzle and some additional extra feature that uses inheritance in some, reasonable way.

Enhancing the puzzle mechanism involving inheritance and math problem-solving in a text adventure game enriches player engagement and demonstrates object-oriented programming's flexibility. Here's a streamlined approach to integrating this puzzle:

- **NPC Interactions for Clues:** Players gather clues through dialogues with NPCs. These clues, framed as math problems, guide players towards locating essential items (T-card, Cheat Sheet, Lucky Pen) critical for game progression.
- **Math Problems as Keys:** Solving math problems uncovers codes to unlock these items. This approach seamlessly blends narrative depth with interactive problem-solving.
- The T-card grants access to Robarts Library (Location 2) and Goldring (Location 8), where NPC 'Akram' offers further hints.
- The Exam Centre remains locked until all items are collected at the dorm room, setting a clear objective for winning the game.

Player position: (0, 0) Moves Left: (50/50)

LOCATION 1

You are currently in your dorm room. In just a few hours you will have to write the hardest exam of your life (MAT157)! But... There's a problem. You misplaced all your things! There's no way you can write the test without your T-card, Lucky pen, and cheat sheet. There's not much time left, you better get going and find your things!

What to do? (type in 'menu' for a list of available commands and 'help [command]' to find how to use a command)

Enter action: *?* go east

Player position: (1, 0) Moves Left: (49/50)

Access Denied: This location needs a T-card to enter.

Player position: (2, 2) Moves Left: (44/50)

LOCATION 7

The Myhal Centre is where you met up with some friends after class yesterday. They were also on that finals week grind, so you didn't stay for long! Oh well, at least you got to socialize for once!

What to do? (type in 'menu' for a list of available commands and 'help [command]' to find how to use a command)

Enter action: *?* go east

Player position: (3, 2) Moves Left: (43/50)

Access Denied: This location needs a T-card to enter.