

TextQuest

I want to create a basic text-based adventure that takes user input to control their character's interaction with the game's environment. The primary focus on the game will be room traversal, inventory management and NPC interaction.

Data Structures:

Linked lists for game map traversal

Each node is effectively a Room class within an area of the game with four pointers and three values

- String this.name
- String this.description
- Inventory this.itemList (explained with the next utilized data structure)
- Room this.north
- Room this.south
- Room this.east
- Room this.west

The player, a subclass of the Guy class (Java won't let me use a "Character" class), should be able to get to every room only by traversing through adjacent rooms.

Hash table: Inventory management for characters and rooms

Every "Guy" and Room has an Inventory of Item objects (each belonging to a subclass of Item like "Weapon" or "Key"). Items in a room can be picked up by the player, which removes them from the room's inventory and adds them to the player's inventory. Items can also be dropped from a player's inventory, which adds them to the room's inventory.

A hash table seems like the best choice for this to ensure $O(1)$ access without relying on index numbers as well as resizing.

Array (Strings): NPC conversations

Each NPC can have a statically sized list of phrases they say in response to player inputs. They can be accessed in $O(1)$. We don't need the complexity of a symbol table, as we can just access the lines of conversation we need with their associated index values.

By the end of the project, users will be able to go through a short adventure (or die in the process).