Programming a single-user dungeon

April 13, 2012

Multi user dungeons

For a long time, playing role-playing games on the computer meant connecting to a remote server and interacting via text with the virtual world. These games, called MUDs (multi user dungeon), were created in the 70s and quickly spread among university students. They are also considered the embryo of famous multi player games as Ultima Online, Everquest and World of Warcraft. A more complete description of its history can be found on Wikipedia (`http://en.wikipedia.org/wiki/MUD`).

A MUD is a mixture of game and chat, with a text-based interface. This allows a very rich description of the world, only limited by the designer's imagination. This interface consists of a prompt, where commands (verbs) are typed and results are output on the screen. There are verbs for moving ("`go [direction]`"), looking around ("`look`") and also to act on items of the world. For example, "`eat apple`" would apply the verb `eat` to the object of name `apple`. Some of the verbs are actually shortcuts. One can use directions (e.g., `north`, `south`, `east` and `west`) as verbs. The rooms in this virtual world are connected by directions, and each has a description of how it looks like.

Project

For this project, you will implement a "single user dungeon" (SUD). This is the single player version of a MUD. The goal is to have a world composed of rooms, each containing a description (how it looks) and linked to other rooms (to allow moving). A room should also keep track of the items in it. The description of the room and which items are in it should appear when the command "look" is used. Items must have at least a name, description and respond to verbs. For example, a guitar that is "used" (i.e., `use guitar`) should play a song. If you look at an item (`look guitar`), you should see a description of how it looks like. You also need to keep track of the player: his name and belongings.

The simplest way to manage this is to create classes Room, Item and Player. Which variables and functions should they contain? A suggestion to make your life simpler: define methods for each verb and call them accordingly. As not every item is the same (some will be edible, others will be playable), you can create classes derived from Item: Food, Drink, MusicalInstrument, Notebook, etc, which answer to different verbs, or even behave differently when `look`'ed. Be creative. You can also play some MUDs for inspiration (links on Wikipedia).

Notes and Hints

1. Try to start with a simple script which just asks for the player's name and allows him to type commands in a prompt. For this, you can loop reading the raw input and split the string in order to get the first word.

2. After you have a working prompt, you can create a single room and have items on it. This should allow you to test different verbs.

3. As the player has belongings, you should implement verbs "pick" and "drop", so he can move an item from the world to his inventory and vice-versa. You should also create a verb (`inventory`) to list his items.

4. If everything is working so far, start linking rooms and test movement. Remember that if room A is north of room B, it also means that room B is south of room A.

5. Command parsing should understand a lazy player. That is, typing just the letter `l` should work the same as `look` and the directions can be: `n`, `s`, `e`, `w`.

6. How about creating an item that is shy, and screams at the player if it's looked at?

7. Suggestion of worlds: a museum which describes all the Nobel prize winners from Göttingen; the inner city of Göttingen or some fantasy place from a book that you like.

8. Some suggestion of verbs: `open`, `press`, `destroy`, `kick`, `eat`, `play`, `read`, `scribble`, `paint`. Invent some more and have fun!