The Dungeon of Dooom Game Specification

February 18, 2013

The Dungeon of Dooom is played on a rectangular grid (the Dungeon) on which the player can move and pick up items. The goal is to collect enough gold and make it to the exit.

1 The Dungeon

The dungeon is made out of square tiles. A tile can be:

Floor Allows a player to walk over it, some may also contain gold. Looks like a dot.

Wall Prevents a player from moving though it. Looks like a hash sign.

Exit A special floor tile necessary for winning the game. Looks like a letter E.

A dungeon can be of arbitrary size so long as it is rectangular. A dungeon should contain at least as much gold as is required to win, and at least one exit tile. Note that exit tiles will never contain gold. A tile with gold on it looks like a letter G.

The map should be redrawn after each action that changes it. A player's location should be shown by the letter P (the player should be shown instead of any possible gold or exit). If there is a bot, it should be shown with a B. Players and bots are not permanent changes to the map – if they move over an exit and the game does not end, it should still be an exit when the player moves away.

2 Setup

You start the game with no gold, and at a random location within the dungeon. This position may contain gold (if you are lucky), may be an empty tile, or it may be an exit tile. You should not be placed inside a wall. The same should be true for your bot if you have one.

3 Winning the Game

The objective of the game is to collect at least a certain amount of gold and then move onto an exit tile in the dungeon. This target amount of gold is different from map to map, and may be as low as zero. If you have enough gold and land on the exit, you should automatically leave the dungeon and the game should finish. The same should go for your bot.

4 Commands

Your program MUST accept the following commands, when used through the command line. These comprise the name of the command (a human readable string) followed by a space and any arguments to the command, then a new line character.

4.1 HELLO

Command: `HELLO`
Response: `GOLD <number>`
The amount of gold required to win the game.

4.2 MOVE

Command: `MOVE <direction>`
Response: `SUCCESS` or `FAIL`
Move one square in the indicated direction. The direction MUST be either `N`, `S`, `E` or `W`.

4.3 PICKUP

Command: `PICKUP`
Response: `SUCCESS, GOLD COINS: <number>` or `FAIL`
To pick up the item in the player's current location. On success, returns the new total of gold in bag.

4.4 LOOK

Command: `LOOK`
Response: `LOOKREPLY <response>`
This action reveals the map around you, showing walls, objects, exit tiles and other players. Right now this command is only for the bot to use, as the command line interface lets you see the whole dungeon. Optionally you can allow the player to call this command for ease of testing. It has no impact on the dungeon or the character. The response should show your field of vision as follows:

```
X???X
?????
??P??
?????
X???X
```

Where each `?` indicates a visible square, `X` indicates squares outside of the field of view, and the `P` indicates the position of the player (who used `LOOK`). The player is always assumed to be at the centre of the reply and is not included explicitly, so as not to obscure the contents of the square in which they are standing. (Note that it is possible to "see through" walls if you are standing directly next to them).

4.4.1 LOOK Example

This is an example of a look reply:

```
LOOKREPLY
X###X
#.###
.GG..
###.#
X#G.X
```

In this example, the player is located on a piece of gold with another gold to the west, a blank square to the east, and walls north and south. This look reply consists of five lines, each terminated by a new line character.

4.5 QUIT

Command: `QUIT`
Response: Game finishes.

There may be no way to win the game if your bot has collected all the gold. Of course, future versions of the game may offer a way out of this conundrum.