***Game overview***

The Dungeon of Doom is played on a rectangular grid, which serves as the game’s board. A human player, acting as a brave fortune-hunter, can move and pick up gold. The goal is to get enough gold to meet a win condition and then exit the dungeon. The player sends a command and (if the command is successful) an action takes place. A full list of the available commands, the game protocol, is on the next page.

The game ends when the human player has collected enough gold and calls the EXIT command on the exit square.

***Board representation***

The dungeon is made up of square tiles. Each tile can be:

1. Player: This tile represents a human player. It is displayed as the letter P.
2. Bot: This tile represents a computer-controlled player, or ‘bot’. It is displayed as the letter B.
3. Empty Floor: Allows a player to walk over it, some may also contain items such as gold. If empty, it is displayed as a period (.)
4. Gold: A special floor tile which allows a player to walk over it and pickup the gold in it. If (and only if) the human player pickups the gold, then the tile is converted into an empty floor tile. It is displayed as the letter G.
5. Exit: A special floor tile that the human player can use to exit the dungeon and win the game by using the QUIT command. It is displayed as the letter E.
6. Wall: Blocks a player from moving through it. It is displayed as a hash sign (#)

***Set-up***

The human player starts the game with no gold, and at a random location within the dungeon.

This position must not contain any gold, but it may be an exit tile. The players must not be placed inside a wall.

***Game Protocol commands***

Your software must allow players to use commands from the ‘game protocol’ (below) on their turns and see the response to those commands. These commands will be entered through the command line. Commands may be upper or lower case. Note: you should not include <and >, these denote where a value should be inserted.

HELLO

The response displays the total amount of gold required for the player to be eligible to win. This number should not decrease as gold as collected.

Example format: Gold to win: <number>

GOLD

The response displays the current gold owned. Example format: Gold owned: <number>

PICKUP

Picks up the gold on the player’s current location. The response is Success and the amount of gold that the player has after picking up the gold on the square. If there is no gold on the square, the response is Fail and the amount of gold that the player had before attempting PICKUP. Example format is: Success. Gold owned: <number>

MOVE <direction>

Moves the player one square in the indicated direction. The direction must be either N, S, E or W. For example, MOVE S. Players cannot move into walls. The response should be either Success or Fail depending on whether the move was successful or not.

LOOK

The response is a 5x5 grid, showing the map around the player. The grid should show walls, empty tiles, gold, exits, and players with the relevant character or symbol. The calling player must be shown at the centre of the grid with a P (human) or B (bot). Visible areas outside of the map should be shown as a wall (‘#’).

The players must not be able to view the map other than by using the LOOK command.

QUIT

Quits the game. If the player is standing on the exit tile E and owns enough gold to win, the response is WIN, followed by an optional winning message. Otherwise, the response is LOSE and quits the game, losing all progress.

Once a command has been entered, the response should be printed and the turn is over

***ADDITIONAL WORK***

***Advanced Feature – BOT***

You can choose to implement a basic computer-controlled player or ‘bot’ to compete against the human player. It is suggested that this be implemented in a class called BotPlayer, which returns commands to the GameLogic in the same way that HumanPlayer does. How the bot determines which commands to use on its turn is up to you:

* A minimum implementation will involve the bot randomly moving around the map.
* A more advanced implementation will have the bot looking for and attempting to chase the human player.
* You can also choose to make your bot a Looter! Our villain is no longer content to just chase the hero, they want to steal the gold too! The Looter bot can pickup gold using the PICKUP command. If it gathers enough gold to win, it can go to an exit tile and leave with the EXIT command. It can still win by catching the human player. As it is a player, the bot should only be able to view the map through the LOOK command, and should only use commands on its turn.