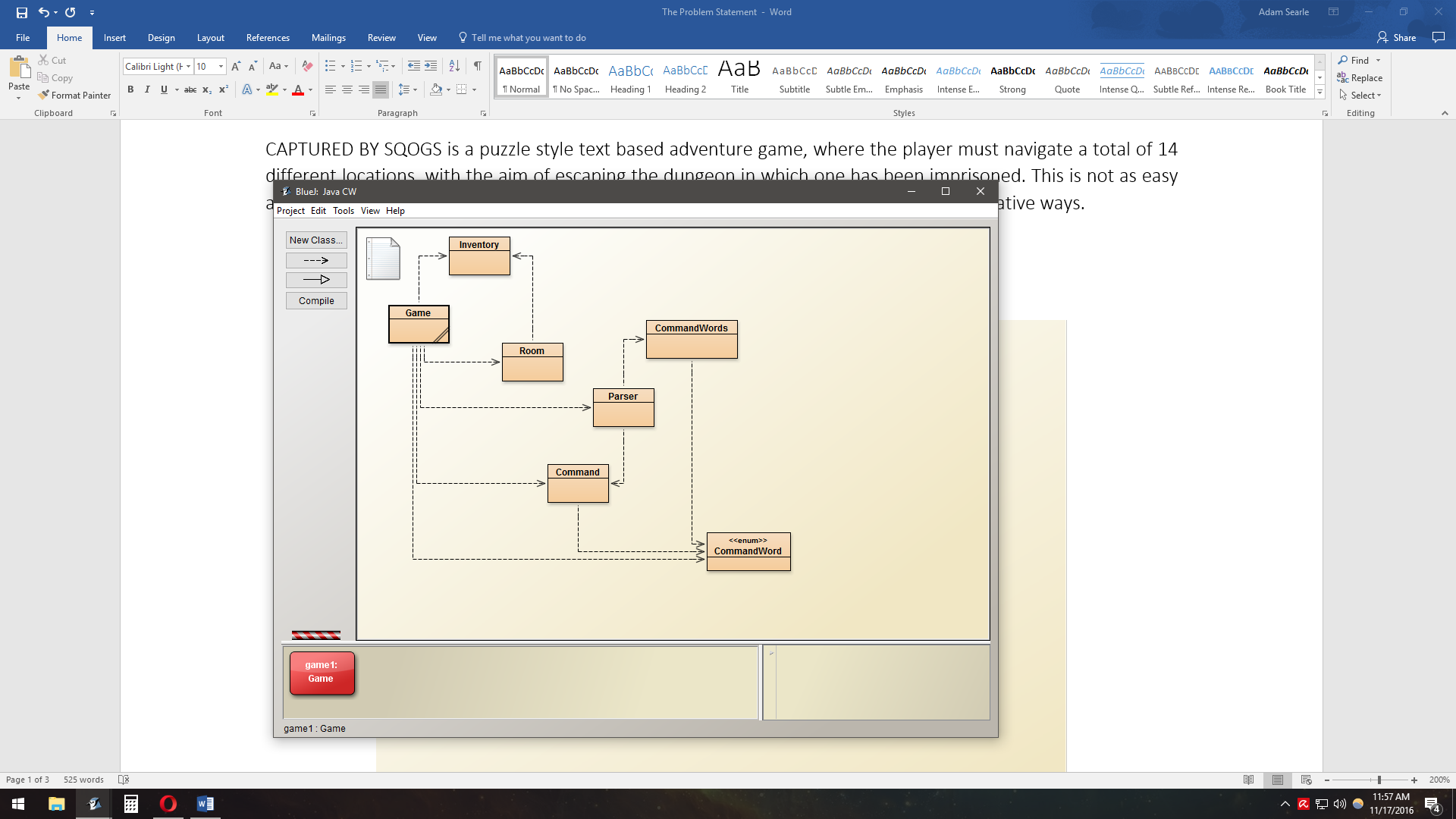
OOP Coursework 1

The Problem Statement

CAPTURED BY SQOGS is a puzzle style text based adventure game, where the player must navigate a total of 14 different locations, with the aim of escaping the dungeon in which one has been imprisoned. This is not as easy as it may first seem, there are various locks, switches and items which must be used in creative ways.

BlueJ Class Diagram



Modifications

I have modified the original program as follows. An Inventory class has been added with the following methods:

addItems ­- Adds a list of passed items to the inventory a list is used to be more flexible.

listItems – Prints a list of items in the inventory.

getItemList – Returns an ArrayList of the inventory items.

containsItem – Takes a string and checks if the inventory contains it.

CommandWord class has been modified to include the words search, use and inventory.

Game class has been modified as follows: Room variable initialization has been moved outside of the createRooms method to allow access outside of the method. 14 new rooms have been added and old ones removed. Additional information added to createRooms method to set up various locks, switches and items. Methods for which are in Room class. When using ‘go’ checks have been added to see if exit is locked and to see if it can be unlocked. Outcomes of additional command words have been added. Mostly these are for ‘use’ as there are various events that take place when the command ‘use’ followed by a specific item in a specific room is entered.

Room class has been modified, adding new HashMaps and lists, also the following methods have been added:

setSearched – to set if a room has been searched yet.

searched – to see if a room has been searched yet.

addSwitch – for adding a switch to the room.

checkSwitch – for checking if a room has any switches.

getSwitches – returns switches and the rooms/exits they are linked to.

getSwitchRoom – returns the room the specified switch is linked to.

getSwitchExit – returns which exit a switch is linked to, given a room.

addItemUse – add the allowing of an item to be used in the room.

getItemUse – check if an item can be used in the room.

setItemUse – for changing if an item can be used in the room.

addItem – for adding an item to the room.

getItem – returns the specified item if it exists.

searchRoom – returns a list of items in the room.

emptyRoom – removes all items from the room.

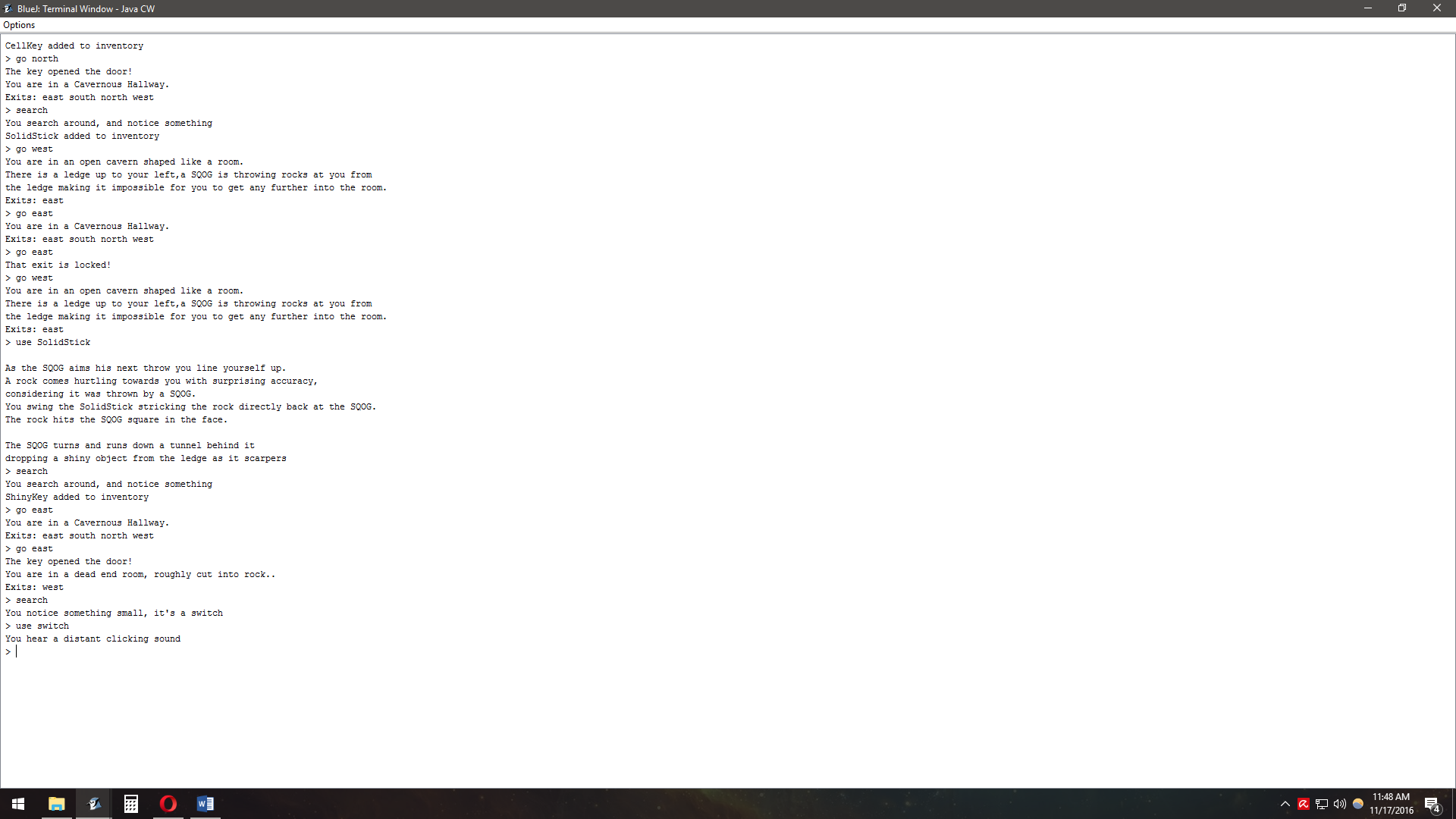
setLockItem – set/link the item required to unlock an exit.

getUnlock – to check if the item required to unlock an exit is in the inventory.

getExitStatus – to check if an exit is locked/unlocked.

setExitStatus – set if an exit is locked/unlocked.

toggleExitStatus – toggle if an exit is locked/unlocked (used for switches, different from above)

On the right you can see many of the main methods in action and below a few more. For example, on the right, we start in a locked cell, search, find a key and now we can unlock the exit. We can also see the use of items and switches. There are many more examples of these implementations at work as the game progresses, some more complex than others.

