Prog2007

Assessment 2

Adventure Game

Kaue Macruz

Student Id: 23771460

1. Game Description

This game is a funny game where the player is out of energy at his house, there is an important soccer game that our actor would like to watch, and his dog is trapped at the basement and needs to go outside to poop.

The game will start in 15 minutes.

The steps of the game are: open the front door, find a flashlight, batteries for the flashlight, find a new fuse to replace the broken fuse in the fuse box located in the kitchen, find the keys for the basement to release the dog, take the dog outside, get a beer in the fridge at the kitchen, go to the tv room, pick up the tv remote control and turn on the tv before the games starts.

The player needs to collect the above items and drop them when necessary since the player is only able to carry 5 items in his inventory.

These are the items available:

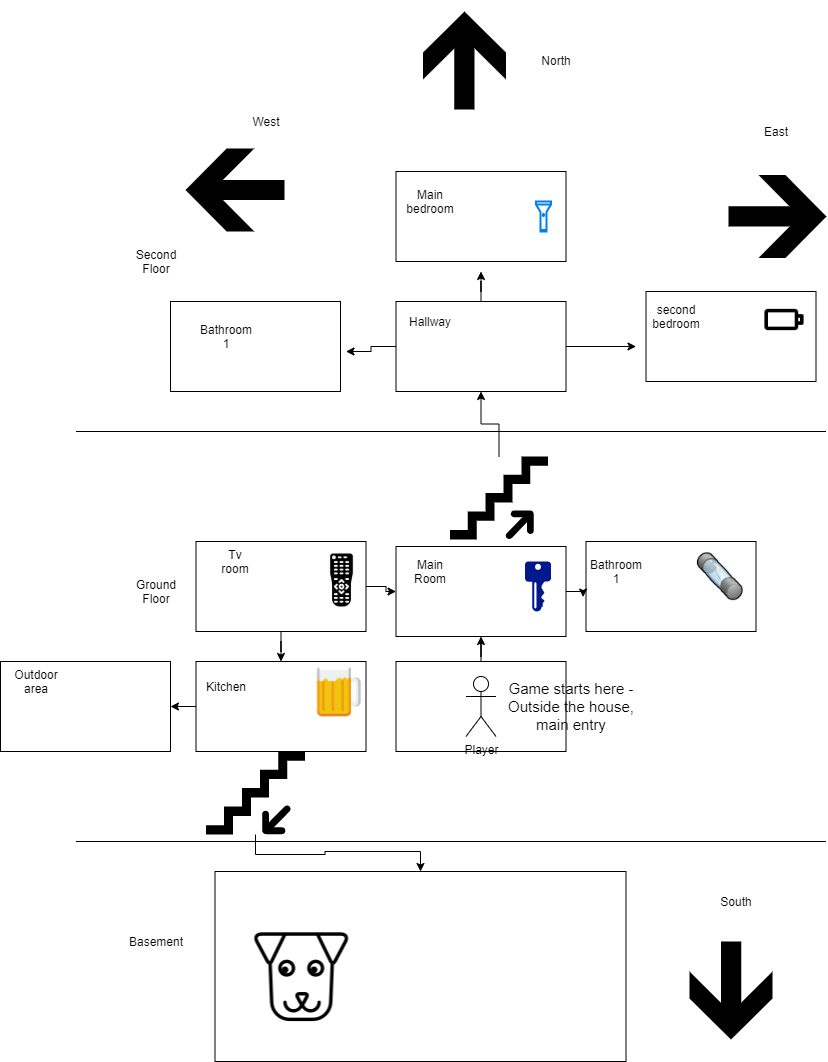
* Flashlight
* Batteries
* Fuse
* Key
* Key1
* Dog
* Beer
* Remote Control

To win the game the player must turn on the tv before the game starts (10 min) and after completing all the tasks previously defined.

1. Game Scenario

Following, you can see the map of the house with the directions as a guide of how to navigate from one room to another, the location of the stairs to go up to the second floor and another do go down to the basement.

In the map you can also see that the flashlight is located in the main bedroom, the battery for the flashlight is located in the second bathroom, the fuse for the fuse box is located in the bathroom 1, the key to open the basement is located in the main room, the dog is located in the basement, the beer is located in the kitchen and the remote control is located in the tv room.



Bathroom 22

Fuse Box

1. Fixes to be made to the zuul-bad game

Class Room Instance variables

• Add an instance variables of type Room for the upstairs exit

• Add an instance variables of type Room for the downstairs exit

setExits method

• Add a parameter of type Room to the setExits method for the upstairs exit

• Add a parameter of type Room to the setExits method for the downstairs exit

• Add an if-statement to the setExits method to assign the up parameter to the upstairs instance variable

• Add an if-statement to the setExits method to assign the down parameter to the downstairs instance variable

getDescription method

• no changes needed

Class Game Instance variables

• Add one array list for each of the rooms with the items located in each room

• Add an array list as the players inventory, max of 5

createRooms method

• Replace rooms’ names that apply to my game

• Add new rooms

• Set the exits for each room

play method

• no changes needed

printWelcome method

• The printWelcome method and the goRoom methods contain code duplication and both print a description of the current room and a list of the exits. Create a new method to handle the description and exits so it will only be written once instead of using the same code twice

• Change text to be printed that fits my game

processCommand method

• Add new command words like: ‘take’ to pick up item, ‘drop’ to drop item, ‘look’ to see a full description of the room, ‘inventory’ too see player’s inventory, ‘use’ to use key to open locked door, to use battery on flashlight, use fuse on fuse box, and to use the remote control and finish the game to turn on the tv and finish the game.

printHelp method

• Replace text to be printed

• Add new words to be used in the processCommand method

• Add ‘items’ command that shows all items in the room

• Add ‘inventory’ command that shows all items the player is carrying

goRoom method

• The printWelcome method and the goRoom methods contain code duplication and both print a description of the current room and a list of the exits. Create a new method to handle the description and exits so it will only be written once instead of using the same code twice

quit method

• no changes needed

1. Duplication Fixes

The printWelcome method and the goRoom methods contain code duplication and both print a description of the current room and a list of the exits. This is not a good design. Ideally, a new method should be written that print a description of the room and and list of the exits, and this method should be called by the printWelcome method and the goRoom method. This way the method is only written once and can be used inside other method making our code cleaner, smaller and more efficient.

This method will be called getRoomExitsAndDescription.

1. Method in the proper class

In the previous step we created a method called getRoomExitsAndDescription in the Game Class, ideally a class of methods should be created to handle all methods that can be used by more than one method. This way it is easier to identify and find the methods, making the classes less messy and easier to understand. We could put every single method in the Game Class, but that would be an extensive and messy class, when you divide the methods in different classes, it makes it much easier for the programmer to navigate through the lines of code and change them if needed.

1. Changes to be made to the zuul-bad game to suit my game

Besides all changes to fix the game previously detailed in step 3, many methods and a couple of classes must be added:

• getLongDescription: to give the description of the room and its exit options

• getExitString: gives a string with the exits available in the room

• timer: to count down the 15 min mark to finish the game

• finishGame: when the player achieves all tasks, a message should be displayed

• takeItem: method that pick up an item located in the room

• dropItem: method that drops item in the current room

• inventory: Shows all items the player is carrying

• look: to see items’ names existing in the room

• StopWatch class: class with methods to give a time to finish the game

• Items class: to handle methods related to the items

1. Improvements

There are many improvements in this game version. In the previous game (zuul-bad game), there was no end of game, and all the player could do was to navigate through the scenario. In my version of the game, the player can pick up items, drop items, use them, look up the description of the room and the items, navigate through the scenario, and there are a few tasks to do in certain order to finish the game. There is also a timer that will set a time to finish the game.