**Algorithm – Haunted House:**

JOptionPane has an option of creating a dropdown menu for selection. I will use this dropdown menu several times to offer options.

Variables (All Private)

* String array of possible backpack objects.
* String for Name.
* Int for backpack item.

**Changes:**

**I had to include two more variables so I’d be able to change the image shown in the Backpack function.**

Int bathroomShower;

If this integer is 0, it means the shower in the Livingroom.

1, it’s the shower in the shared 2nd floor bathroom

2, it’s the shower in the master bathroom.

Boolean livingroomMirror

True, the mirror in the livingroom bathroom

False, the mirror in the shared 2nd floor bathroom.

Public Methods

Name Request Method

* Just a setting method
* Ask for name. Store in string for name, private data.

One method for these rooms, including Front door and Stairs, no return type no arguments

* Asks inputs and contains if statements depending on input using JOptionPane.
* Front Door
* Living Room
* Bathroom Living Room
* Dining Room
* Pantry
* Master Bedroom
* Master Bathroom

One method for stairs room, no return type, one integer argument to indicate entry point

* Stairs Method
* Int argument to signify entry point (0 for Downstairs, 1 for Bedroom 1, 2 for Bedroom 2)
* Different if branches depending on entrypoint int.
* If from downstairs: Options are Bedroom1, Bedroom2, Master Bedroom.
* If from Bedroom1: Options are Master Bedroom, Bedroom 2, Stairs.
* If from Bedroom2: Options are Master Bedroom, Bedroom 1, Stairs.

Two methods for bedroom 1 and bedroom 2, no return type, one Boolean argument to indicate entry point

* Bedroom 1 Method
* Boolean Argument to signify coming from stairs
* False, If from bathroom: Options are Stairs, Rocking Chair, Window
* True, If from Stairs: Options are Bathroom, Rocking Chair, Window
* Bedroom 2 Method
* Boolean Argument to signify coming from stairs
* False, If from bathroom: Options are Stairs, Doll House, Dresser
* True, If from Stairs: Options are Bathroom, Doll House, Dresser

One method for bathroom, using an integer to indicate bathroom location or entry point.

* Bathroom Method
  + 0, if First floor bathroom
    - Options are: Mirror, Shower
  + 1 if 2nd Floor Bathroom from Bedroom 1
    - Options are: Bedroom 2, Mirror, Shower
  + 2 If 2nd Floor Bathroom from Bedroom 2
    - Options are: Bedroom 1, Mirror, Shower

Backpack Method, no return type no arguments

* Contains output statements contained in if statements, depending on backpack integer in private data.

Changes:

One new method, chosenDestination.

My oversight previously did not realize that JOptionPane’s returns would return strings rather than the index of the chosen option from the passed in array. I created this method to rectify this.

This method returns an integer, after being passed in the return of JOptionPane(the string option that was chosen) and the array of options JOptionPane used. This integer return is the index.

This index is then used in the switch statement of each function where options are offered.

First, check if location is null and if so, return -1; so program will terminate at default switch case.

If not null,

For the length of the passed in array, compare each array value to the chosen option. If equal, return that index and break statement.

The JOptionPane Dropdown menu uses an array of strings to offer options. For each dropdown menu, a string array of what the options will say must be created. These are used as the arguments for creating the JOptionPane object.

* Front Door – no return, no arguments
  + Create input int.
  + Create String Array of Options: Stairs, Dining Room, Living Room
  + JOptionPane, Dropdown menu with image, set return to input int.
  + IF - Input is 0 for Stairs
    - * Call Stairs method
    - If Else – Input is 1 for Dining Room
      * Call Dining Room Method
    - If Else – Input is 2 for Living Room
      * Call Living Room Method
    - Else
      * Print to console: Program terminated.
* Living Room – no return, no arguments
  + Create input int
  + Create String Array of Options: Bathroom, Chest
  + JOptionPane, Dropdown menu with image, set return to input int.
  + If – Input is 0 for Bathroom Living Room
    - * Call Bathroom with argument 0
    - If Else – Input is 1 for Explore: Chest
      * Set backpack item int in private data to index of backpack string array for Chest
      * Call Backpack Method
    - Else
      * Print to console: Program terminated.
* Dining Room – no return, no arguments
  + Create input int
  + Create String Array of Options: Kitchen, Candelabra
  + JOption Pane, Dropdown menu with image, set return to input int
  + If – input is 0 for Kitchen
    - * Call Kitchen Method
    - If Else – input is 1 for Candelabra
      * Set backpack item int in private data to index of Candelabra in backpack string array
      * Call Backpack Method
    - Else
      * Print to console: Program terminated.
* Kitchen – no return, no arguments
  + Create input int
  + Create string array of options: Pantry, Refrigerator, Cabinet
  + JOptionPane, dropdown menu with image, set return to input int
  + If – input is 0 for Pantry
    - * Call Pantry Method
    - If else – input is 1 for Refrigerator
      * Set backpack item int in private data to index of Refrigerator in backpack string array
      * Call Backpack method
    - If Else – input in 2 for cabinet
      * Set backpack item int in private data to index of Cabinet in backpack string array
      * Call backpack method
    - Else
      * Print to console: Program terminated.
* Pantry – no return, no arguments
  + Create input int
  + Create String Array of Options: Dusty recipe box, Broom
  + JOptionPane, Dropdown menu with image, set return to input int
  + If – input is 0 for Dusty recipe box
    - * Set backpack item int in private data to index of Dusty recipe box in backpack string array
      * Call Backpack method
    - If else – input is 1 for Broom
      * Set backpack item int in private data to index of Broom in backpack string array
      * Call backpack method
    - Else
      * Print to console: Program terminated.
* Stairs – no return, int argument named location
  + Create input int
  + Argument set to int type named entryPoint
  + If entryPoint = 0, coming from downstairs
    - Create string array of options: Master Bedroom, Bedroom 1, Bedroom 2
    - JOption Pane, Dropdown menu with image, set return to input int
    - If input is 0, Master Bedroom
      * + Call Master Bedroom Method
      * If Else input is 1, Bedroom 1,
        + Call Bedroom 1 Method with argument true
      * If Else input is 2, Bedroom 2,
        + Call Bedroom 2 Method with argument true
  + If Else entryPoint = 1, coming from bedroom 1
    - Create string array of options: Master Bedroom, Stairs, Bedroom 2
    - JOption Pane, Dropdown menu with image, set return to input int
    - If input is 0, Master Bedroom
      * + Call Master Bedroom Method
      * If Else input is 1, Go down stairs,
        + Call Front Door
      * If Else input is 2, Bedroom 2,
        + Call Bedroom 2 Method with argument false
  + If Else entryPoint =2, coming from Bedroom 2
    - Create string array of options: Master Bedroom, Stairs, Bedroom 1
    - JOption Pane, Dropdown menu with image, set return to input int
    - If input is 0, Master Bedroom
      * Call Master Bedroom Method
        + If else input is 1, Go down stairs

Call front door method

* + - * + If else input is 2, bedroom 1,

Call bedroom 1 method with argument false

* + - * + Else

Output to console: Program terminated

* + Else
    - Print to console: Invalid argument. Program terminated.
* Bedroom 1 – no return, Boolean argument for Location
  + Create input int
  + Argument set to Boolean type named FromStairs
    - If fromStairs = true
      * Create string array of options: Bathroom, Rocking Chair, Window
      * JOption Pane, Dropdown menu with image, set return to input int
      * If – input is 0 for Bathroom

Call Bathroom Bedroom with argument int 1

* + - * + If else – input is 1 for Rocking Chair

Set backpack item int in private data to index of Rocking chair in backpack string array

Call Backpack Method

* + - * + If Else – input is 2 for Window

Set backpack item int in private data to index of Window in backpack string array

Call Backpack Method

* + - * + Else

Print to console: Program terminated.

* + - If fromStairs = false
      * Create string array of options: Stairs, Rocking Chair, Window
      * JOption Pane, Dropdown menu with image, set return to input int
      * If – input is 0 for Stairs

Call Stairs with argument 1

* + - * + If else – input is 1 for Rocking Chair

Set backpack item int in private data to index of Rocking chair in backpack string array

Call Backpack Method

* + - * + If Else – input is 2 for Window

Set backpack item int in private data to index of Window in backpack string array

Call Backpack Method

* + - * + Else

Print to console: Program terminated.

* Bedroom 2 – no return, Boolean argument for Location
  + Create input int
  + Argument set to Boolean type named FromStairs
    - If fromStairs = true
      * Create string array of options: Bathroom, Doll House, Dresser
      * JOption Pane, Dropdown menu with image, set return to input int
      * If – input is 0 for Bathroom

Call Bathroom Bedroom with argument int 2

* + - * + If else – input is 1 for Doll House

Set backpack item int in private data to index of Doll House in backpack string array

Call Backpack Method

* + - * + If Else – input is 2 for Dresser

Set backpack item int in private data to index of Dresser in backpack string array

Call Backpack Method

* + - * + Else

Print to console: Program terminated.

* + - If fromStairs = false
      * Create string array of options: Stairs, Doll House, Dresser
      * JOption Pane, Dropdown menu with image, set return to input int
      * If – input is 0 for Stairs

Call Stairs with argument 2

* + - * + If else – input is 1 for Doll House

Set backpack item int in private data to index of Doll House in backpack string array

Call Backpack Method

* + - * + If Else – input is 2 for Dresser

Set backpack item int in private data to index of Dresser in backpack string array

Call Backpack Method

* + - * + Else

Print to console: Program terminated.

* Bathroom – no return, int argument named Location
  + Create input int
  + Argument set to int type named entryPoint
    - If entryPoint = 0, bathroom at living room
      * Create string array of options: Mirror, Shower
      * JOption Pane, Dropdown menu with image, set return to input int
      * If – input 0 is Mirror

Set backpack item int in private data to index of Mirror in backpack string array

* + - * If Else - input is 1 for Shower
        + Set backpack item int in private data to index of backpack string array for Shower
        + Call Backpack Method
      * Else
        + Print to console: Program terminated.
    - If else entryPoint = 1, bathroom at 2nd floor from Bedroom 1
      * Create string array of options: Bedroom 2, Mirror, Shower
      * JOption Pane, Dropdown menu with image, set return to input int
      * If input 0, Bedroom 2

Call Bedroom 2 with argument false

* + - * + If Else – input 1 is Mirror

Set backpack item int in private data to index of Mirror in backpack string array

Call Backpack Method

* + - * + If Else - input is 2 for Shower

Set backpack item int in private data to index of Shower in backpack string array

Call Backpack Method

* + - * Else
        + Print to console: Program terminated.
        + Print to console: Program terminated.
    - If else entryPoint = 2, bathroom at 2nd floor from Bedroom 2
      * Create string array of options: Bedroom 1, Mirror, Shower
      * JOption Pane, Dropdown menu with image, set return to input int
      * If input 0, Bedroom 1

Call Bedroom 1 with argument false

* + - * + If Else – input 1 is Mirror

Set backpack item int in private data to index of Mirror in backpack string array

* + - * + If Else - input is 2 for Shower

Set backpack item int in private data to index of Shower in backpack string array

Call Backpack Method

* + - * Else
        + Print to console: Program terminated.
    - Else
      * Print to console: Invalid argument. Program terminated.
* Master Bedroom – no return, no arguments
  + Create input int
  + Create string array of options: Master Bathroom, Jewelry Box, Intricate Oil Lamp
  + JOption Pane, Dropdown menu with image, set return to input int
  + If input = 0, Master Bathroom
    - * Call Master Bathroom Method
    - If else input = 1, Jewelry Box
      * Set backpack item int in private data to index of Jewelry Box in backpack string array
      * Call Backpack Method
    - Else
      * Print to console: Program terminated.
* Master Bathroom – no return, no arguments
  + Create input int
  + Create string array of options: Intricate Oil Lamp, Shower
  + JOption Pane, Dropdown menu with image, set return to input int
  + If input = 0, Intricate Oil Lamp
    - * Set backpack item int in private data to index of Intricate Oil Lamp in backpack string array
      * Call Backpack Method
    - If Else = 1, Shower
      * Set backpack item int in private data to index of Shower(2) in backpack string array
      * Call Backpack Method
    - Else
      * Print to console: Program terminated.
* Backpack Method – no return, no arguments
  + Series of if else statements depending on Backpack int from private data, only outputting statement.
  + If 0 –
    - * Output: Backpack is empty. This shouldn’t be able to be reached.
    - If else, backpack item is 1
      * Output: Chest in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 2
      * Output: Candelabra in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 3
      * Output: Refrigerator in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 4
      * Output: Cabinet in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 5
      * Output: Dusty Recipe Box in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 6
      * Output: Broom in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 7
      * Output: Mirror in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 8
      * Output: Shower in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 9
      * Output: Rocking Chair in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 10
      * Output: Window in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 11
      * Output: Doll House in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is 12
      * Output: Dresser in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is Jewelry Box
      * Output: Jewelry Box in Backpack, using backpack array.
      * Output: Death message, Game is over.
    - If else, backpack item is Intricate Oil Lamp
      * Output: Intricate Oil Lamp in Backpack, using backpack array.
      * Output: Death message, Game is over.

Tester Class

As the program iterates through itself and does all inputs,

Program starts at Name Request method. Then each method is called in program.

Tester class will consist of creating an object.

Then running the name request method, which will be renamed to HauntedHouseStart.

All inputs are done in the methods themselves, none are done in the main class.