

Name: Ya Lien

Due Date: 8/7/2016

Final Project Design and Test Plan

Game Description and Walk Through:

Your cat has sneaked into your mean neighbors' house while they are out to dinner. You need to find the cat and take her out of the house before the neighbors return.

It's a Sunday night when you found that your front door is wild open and your cat is missing. You went out to the street to search for her, that's when you hear faint meowing answering from your mean neighbor's house. You checked out the house and are relieved to find that no one is home. You went to the back of the house and found the kitchen door unlocked.

You decided to enter from the kitchen and find your cat before the neighbors return.

In the kitchen you will find a switch to turn on the light. You need the light on to see the inside of the kitchen.

Also in the kitchen, you will find peanut butter

(enter 1 to pick up peanut butter)

Next, you can decide to enter the bedroom, storage, or bathroom.

(Go to the bathroom first)

If enter bathroom, you will be able to open the medicine cabinet and find two pieces of bread.

(enter 1 to pick up the bread)

(Then go to storage)

If enter storage, you will be able to find a hidden passage that leads to the attic. If you choose to open the door, you will find strawberry jam.

(enter 1 to pick up strawberry jam)

(The go to bedroom)

If you enter the bedroom and already have the three items in bag, you can put the 3 items together and make a peanut butter jelly sandwich! Which is your cat's favorite!

(enter 1 to pick up peanut butter jelly sandwich)

Then, you can enter the Attic.

When you are in the Attic, you can hear your cat purring but she refuses to come out under the bed, you need to use the peanut butter sandwich to lure the cat out!

(enter 1 to use)

Grab the cat and put in bag. You and the cat are now safe!

(enter 1 to put in bag)

In this game, moving to a new room is equal to 1 minute passed. The player will have 45 minutes (45 moves) before the neighbors return.

Hint will be provided through each step. If you follow the guide, the game should be finished within 4 minutes.

Design Description:

This program will have 5 sub classes, Kitchen, Bedroom, Bathroom, Storage, and Attic. They are all derived from a base class called Space. The Space class will have 4 Space pointers as its private member variables. It will also have a public move function and a pure virtual function that allows each of the sub class to perform a special action, and a pure virtual function that allows the user to collect different items in different room.

The structure of the Spaces will be changed when the player enters the Storage: a new room, the Attic, will be added to the structure and one of the bedrooms will be removed.

The time limit is implements as 1 move-through-space = 1 minute passed. Time passed will be displayed at the end of each move.

This program will also have a player class to keep track of their current location and have a backpack for the player to carry items around. The player's backpack will be implemented as a linear linked list with a head and tail pointer. The player class will also have a function to keep track of the time passed. The reason why I create an independent class is because I feel like the player's action is not part of the Space, but interaction with the Space.

Pseudocode

Player class

- Keep track of which space they player is in
 - setLocation(spaceName)
 - getLocation()
- Have a container to carry items found in spaces – need to have a limit
 - addToBag() //only allows up to 3 items at a time
 - linked list
 - counter to keep track of the number of items in bag.
 - Remove()
 - isEmpty
- Keep track of the time

- Time counter

Space class

- 4 pointers that link to other spaces
 Space *up;
 Space *left;
 Space *right;
 Space *back;
- Move()
 If up
 Return up;
 If left
 Return left;
 If right
 Return right;
 If back
 Return back;
 (in main:
 Space *where = move();
)
- specialAction() – needs to be pure virtual
- player need to interact with the spaces – ie turn on light, open the door
- need to collect items
- setItem() – pure virtual

Kitchen class

- specialAction()
 - turn on the light
- setItem()
 peanut butter = true
- boolgetItem()

Bathroom class

- specialAction()
 - open the medicine cabinet
- takeItem()

Storage class

- takeItem()
- specialAction()
- open door to a second storage

Bedroom class

- specialAction()
 - put the 3 items together
- 3 items together makes a peanut butter jelly sandwich! Which is your cat's favorite!
 - Use it to lure the cat out of the upstairs bedroom
 - getItem()
 - how do I check if I have all 3 times

Attic class

- specialAction()
 - turn on the light
- find the cat!

Test Plan

Test Kitchen Class	Expected Result	
specialAction() function <ul style="list-style-type: none"> • put in parameter 1, p1 • put in parameter 2, p1 	<ul style="list-style-type: none"> • return true • return false 	✓ ✓
takeItem() function <ul style="list-style-type: none"> • put in parameter 1, true • put in parameter 2, true • put in parameter 1, false • put in parameter 2, false 	<ul style="list-style-type: none"> • print: found peanut butter print: picked up peanut butter • print: found peanut butter print: item not picked up • go to other rooms • go to other rooms 	✓ ✓ ✓ ✓
Test Bathroom Class	Expected Result	
specialAction() function <ul style="list-style-type: none"> • put in parameter 1, p1 • put in parameter 2, p1 	<ul style="list-style-type: none"> • return true • return false 	✓ ✓
takeItem() function <ul style="list-style-type: none"> • put in parameter 1, true • put in parameter 2, true • put in parameter 1, false • put in parameter 2, false 	<ul style="list-style-type: none"> • print: found bread print: picked up bread • print: found bread print: item not picked up • Nothing else in the room • Print: nothing else in the room 	✓ ✓ ✓ ✓

Test Storage Class	Expected Result	
specialAction() function <ul style="list-style-type: none"> put in parameter 1, p1 put in parameter 2, p1 	<ul style="list-style-type: none"> return true return false 	✓ ✓
takeItem() function <ul style="list-style-type: none"> put in parameter 1, true put in parameter 2, true put in parameter 1, false put in parameter 2, false 	<ul style="list-style-type: none"> print: found bread print: picked up bread print: found bread print: item not picked up Nothing else in the room Print: nothing else in the room 	✓ ✓ ✓ ✓
Test Bedroom Class	Expected Result	
specialAction() function <ul style="list-style-type: none"> put in parameter 1, p1 put in parameter 2, p1 	<ul style="list-style-type: none"> return true return false 	✓ ✓
takeItem() function <ul style="list-style-type: none"> put in parameter 1, true put in parameter 2, true put in parameter 1, false put in parameter 2, false 	<ul style="list-style-type: none"> print: found peanut butter jelly sandwich print: picked up peanut butter jelly sandwich print: found peanut butter jellysandwich print: item not picked up Nothing else in the room Print: nothing else in the room 	✓ ✓ ✓ ✓
Test Attic Class	Expected Result	
specialAction() function <ul style="list-style-type: none"> put in parameter 1, p1 put in parameter 2, p1 	<ul style="list-style-type: none"> return true return false 	✓ ✓
takeItem() function <ul style="list-style-type: none"> put in parameter 1, true put in parameter 2, true put in parameter 1, false put in parameter 2, false 	<ul style="list-style-type: none"> print: found your cat print: picked up your cat print: found your cat print: item not picked up Print: nothing else in the room Print: nothing else in the room 	✓ ✓ ✓ ✓