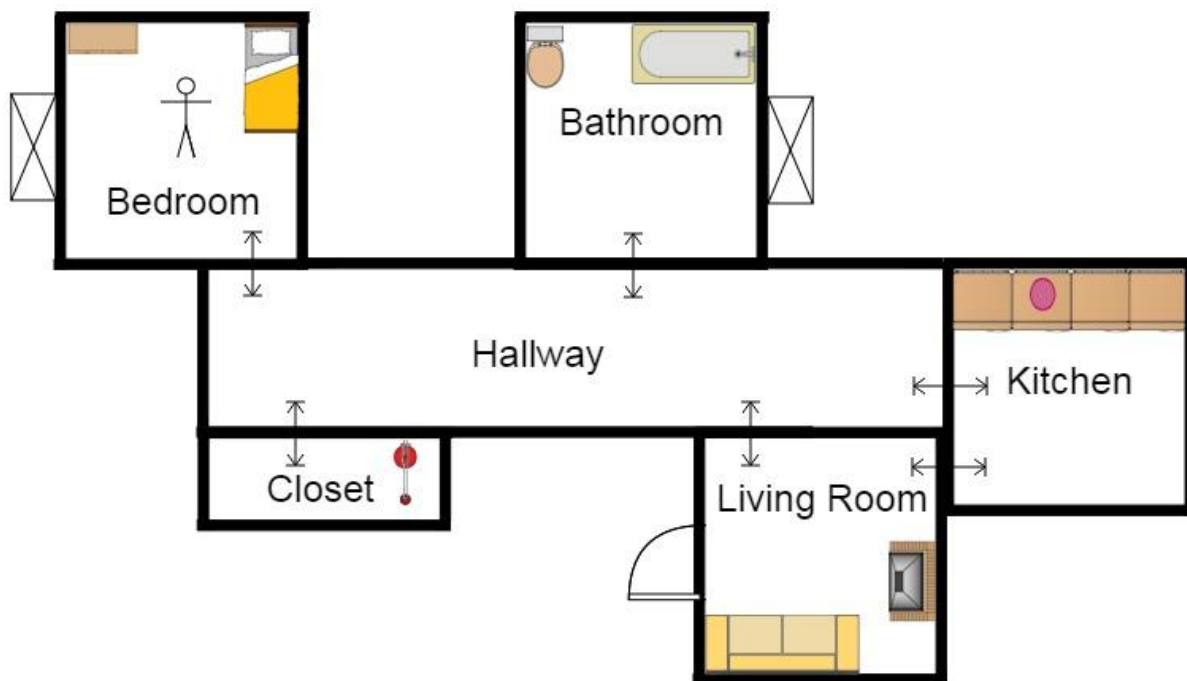


## CS 162 Final Project Reflection

### Design Plan:

For this assignment, I chose to create a scenario where a burglar is trying to rob a house. Unfortunately, he sets off a silent alarm and becomes trapped inside a house with the dog. If he makes too much noise, the dog will attack him and he won't be able to escape in time. If he takes longer than 60 seconds, the police will arrive and he will be sent to jail. The only ways to successfully escape and win the game are to find a key to the front door or a sledgehammer to break the glass on one of the windows.

In order to implement it, I have created a Space parent class and 5 subclasses which will be the rooms of a house. The house consists of a living room, a hallway, a kitchen, a bedroom and a bathroom. Here is a layout of the house:



(The boxes with X's drawn through them represent windows). There is a sledgehammer in the closet and a key inside a tin box on the shelves in the kitchen. Once the alarm is triggered, the only ways out are:

1. Find the sledgehammer in the closet, go to the window in the bedroom or the window in the bathroom, smash the glass and escape
2. Find the key in the kitchen, go to the front door, unlock it and escape.

Class descriptions:

Space class:

- Protected variables:
  - 4 space pointers (to travel to connected spaces)
  - 1 player pointer
- Public functions:
  - Constructor and default constructor
  - Virtual function for presenting menu and options
  - Virtual function for receiving input on where to go next
  - Virtual function for adding connected spaces

Each of the 5 subclasses as a constructor which initialized all of it's connected spaces to null and sets a player pointer and has an option function for presenting that room's specific menu.

Player class:

- Private variables:
  - Bool variable for whether player has found the key
  - Bool variable for whether player has found the sledgehammer
  - Bool variable for whether the player has escaped
  - Backpack container (string array) to hold key and sledgehammer
  - Int count variable to place items in the correct position of the backpack array
- Public functions
  - Default constructor
  - AddItem function which adds either the key or sledgehammer into the backpack
  - Bool GetKey to check if the player has found the key
  - Bool GetSH to check if the player has found the sledgehammer
  - Accessor and mutator function for escaped variable

Main:

- Initialize spaces
- Set current room to bedroom
- Use addSpace to connect the rooms
- Initialize variables for user input, time
- Display game introduction
- Display main menu
  - Start game
  - Show hint
- Start timer, calculate elapsed time
- While loop to continue the game until 1 of 3 things occurs:
  - Dog attacks player
  - Player escapes
  - Time runs out
- Exit messages for above 3 scenarios

## Testing:

I had several issues with coming back from the added items in some of the rooms (i.e. user checks tin, tin is empty - instead of going back to the kitchen menu, the function kept going back to the shelf menu which was redundant. Other than that, my classes and functions behaved as expected. I decided to step through each path that the user could take after starting the game to ensure all functions were working correctly.

### Testing results (starting from bedroom):

- Check window, locked, try to open, dog attack = LOSE
- Check window, locked, leave room, go to hallway, go to bathroom, check window, locked, try to open, dog attack = LOSE
- Leave room, go to bathroom, check window, locked, leave room, go to hallway, go to living room, check front door, locked, leave room, go to kitchen, check shelves, inspect tin, find key, take it, leave room, go to living room, check door, locked, key works = WIN
- Leave room, go to hallway, check closet, take sledgehammer, leave room, go to hallway, go to bedroom, check window, locked, sledgehammer works = WIN
- Leave room, go to hallway, check closet, take sledgehammer, leave room, go to hallway, go to bathroom, check window, locked, sledgehammer works = WIN
- Travel through all rooms until time runs out without escaping, police arrive = LOSE