Map or Multimap?

Final Project

Requirements:

- 5 different spaces of at least 3 different types (derived Space classes)
- Each space will have 4+ pointer variables that link to other spaces
- Space will be an abstract class w/function Special being a pure virtual
  - each type of space will have a special ~~function~~ action
- Programmer defined goal
- Must keep track of where the player is. Map?
- Player will have a backpack (~~array~~ ~~vector~~ vector) with a limited amount of space
  - one or more stored items must be needed for the solution to the game
- Must allow the player to interact w/environment besides picking up items
- Provide a menu option to play the game or see the rules of the game

Space → Beach (dig in sand) Lighthouse (searchlight) Player
investigate
~~Field (grassy knoll)~~ Garden
addItems
Shack (disassemble chair) pry floor board
dropItems (to current space)
Boat (operate boat)
moveLocation
Cave (find gas)
interact

Pocket at start of game

Items (strings)

Backpack storage space = ~~10~~ 4

gasoline (in cave)
map
bandages (in shack)
key (beach) to the lighthouse
~~metal pole~~ (in ~~field~~) garden
~~crowbar~~ note (boat)
note2 (lighthouse)
~~flashlight (shack)~~

Time limit will be water/thirst

Player

Private:

vector<string> backpack; map<spaces> playerMap;

bool interaction, investigation; string dropItem;

Public:

void addItems (String newItem); // first if-else statement, if backpack.size >= 3 cout the backpack ~~is~~ is full, drop an item first. else the passed through string item gets pushed back to the backpack

void displayBackpack (); // loops through and couts the item(s) currently in the backpack

string dropItem (); // makes a call to displayBackpack and then asks users what item they'd like to remove, then removes said item

bool interact (); // this will ~~return~~ change interaction to true and return it

bool investigate (); this will change investigation to true and return it

void moveToLocation (); ~~prompt the user to~~ check where user currently is. Prompt user to select which location to go to. If user is in lighthouse and they have the oar then they have the option to go on the ~~lighthouse~~ cave. If they are in the cave they can go in the lighthouse; In order to move about use the map STL player map

put in game class

current location pointer pointing to currently occupied space, update it to the new space

void setCurrentLocation (Space *OS;)

Space getCurrentLocation ();

GG ~~747~~	697	772
GO ~~664~~	508	505
574 ~~614~~	529	689

3

Space
Protected:
Player pObj; vector <string> itemStorage;
string uniqueItem1, uniqueItem2;
Public:
Space(); constructor
~Space(); destructor
virtual void describePlace(); this function will be called each time the user enters the space and will give a general description of the environment
virtual void revealInvestigation(player obj); this function first tests and checks if the player investigate object returns true. If it returns true then it will give a description of areas that are able to be interacted with (special) and what items are available to be picked up
virtual void displaySpacesItems(); iterates through the vector to display items available in that speece
virtual void addOrRemoveItems(player obj); Takes a player object If player adds an item then the itemStorage for this location will loose its item. If the player drops an item then it will add that item to item Storage
virtual void special() = 0;

1 Shack
2 Garden
3 Beach
4 Lighthouse
5 Boat

shack, garden, lighthouse, boat, beach

Game

Private:

Player obj; Space *obj;

int userSelection;

bool winGame, looseGame;

Public

Game(); constructor

~Game(); destructor

void Menu(); menu item for determining whether the user picks up or drops items, investigates the area, checks their backpack, uses special or moves locations

void gameIntro(); this will ~~display~~ describe the opening scene for the player

~~void~~ bool gameWon(); returns true once player wins

~~void~~ bool gameLoss(); returns true if player loses

Initialize all of the space objects in Game, then set the pointers in the Game constructor