Maliha Syed
CS162

Final Project – Quest to Save Middle-Earth

Design

I. Input Validation:
   a) For integers with min and max
   b) For string, only allows directions “n”, “s”, “e”, “w”

II. Menu:
   a) Show the menu choices
   b) Accept user choice after validation

III. Quest:
   a) Member variables; Space pointer to current location, Hero pointer to hero, Map pointer to map, Menu pointer to menu, integer daysLeft, and bool gameOver
   b) Default constructor that takes in the player’s name
   c) Intro that displays the game’s introductory text to the player and displays the map
   d) ShowMap function that retrieves the map
   e) HeroDead function that checks the energy of the hero
   f) OutOfTime function that checks the days left
   g) CheckVictory function that checks if the all the special items have been collected
   h) Play function that that calls functions for location events, travels to the next location, and manages energy/daysLeft until game over
   i) Destructor function that deletes the hero pointer

IV. Space:
   a) Abstract class with member variables; integer days, strings name and type, and 4 Space pointers (up, down, left, right)
   b) Constructor that sets pointers to null, and virtual destructor
   c) Pure virtual intro and events functions
   d) Getter and setter functions for the 4 Space pointers
   e) Getters for name, type, and days

V. Shire:
   a) Child class of Space
   b) Default constructor that sets name and destructor
   c) Intro function with a text intro
   d) Events function that has 3 different interactions

VI. Kingdom:

a) Child class of Space
b) Default constructor that sets name, type, and days
c) Intro function with a text intro
d) Events function that has 4 different interactions

VII. Forest:

a) Child class of Space
b) Default constructor that sets name, type, and days
c) Intro function with a text intro
d) Events function that has 4 different interactions

VIII. Mountains:

a) Child class of Space
b) Default constructor that sets name, type, and days
c) Intro function with a text intro
d) Events function that has 4 different interactions

IX. Mordor:

a) Child class of Space
b) Default constructor that sets name and type
c) Intro function with a text intro
d) Events function that returns 0

X. Hero:

a) Member variables name, energy, Space pointer to current location, backpack array of 10 Item pointers, and a Map object map
b) Constructor that takes in a name and assigns it to name, sets energy to 10, and sets location to current location.
c) Default destructor that deletes all the Item pointers in the array
d) Getters and setters for name, energy, and location
e) EatFood function that adds to the energy
f) FightOrc function that iterates through the backpack array and deletes a weapon if one is present, or decrements the energy
g) CollectWeapon function that adds a weapon to the array if there is space available
h) CollectItem function that adds the special items to the backpack
i) PrintItems function that prints all the items in the backpack
j) IsDead function that returns true if energy is 0 or less
k) CheckItems function that returns true if all special items are there

XI. Item:

a) Member variable string type
b) Constructor that takes in a string and assigns it to the type
c) Getter that returns the type

XII. Map:

a) Member variables Space pointers to each space and currentLocation
b) Constructor that initializes new Space pointers with the correct name
c) Private function createMap that connects all the locations via the 4 pointers in each space
d) DisplayMap function that outputs the locations in a map form
e) Getter and setter for the current location
f) Destructor that deletes all Space pointers and assigns them to null

XIII. Main:

l) Instantiate a Quest object
m) Run quest simulation

Test Plan

Test Scope	**Description**	**Expected Result & Observed Result**
Menu with validation	Initialize and show Menu object	Displays menu to user
	Returning a menu option; tested with large numbers, other characters, floats, spaces, just enter	Outputs error if user inputs a string where any character is not numerical, or the string is empty, or the number is less than 1 or greater than 2
	Choosing a menu option	If choice is 2, then location is changed, if choice is 1 event functions are called
	chooseDirection()	Displays location choices
intro()	Text output and calling showMap	Text should be outputted followed by the map
play ()	Runs quest simulation	Should correctly call functions and progress through the locations
currentLocation	Stores the current Space location of the hero	Should be correctly updated with the location
events()	Randomly picks an event depending on the type of	Should, according to the rand function, return an

	the location	integer dependent on which type of event is chosen
collectItem()	Collects the special item	Should correctly add the special item to the array
collectWeapon()	Adds weapon to backpack	Should correctly add the weapon to the array if there is space
eatFood()	Increments the energy by 1	Function should correctly add one to the hero's energy
fightOrc()	Uses weapon in the array	Correctly deletes a weapon if one is present in the array, or decrements the energy if not
heroDead ()	Checks if energy is <= 0	Function should set gameOver to true if energy is <= 0
outOfTime()	Checks if daysLeft is <=0	Function should set gameOver to true if daysLeft is <= 0
showDirection()	Menu that shows the directions from current location	Function should correctly display only the locations that can be travelled
chooseDirection()	Menu function that returns n,s,e,w if not pointing to null	Function should correctly return the chosen direction character
printItems()	Outputs the items in the backpack	Function should correctly print out the items in the array that don't point to null
setLocation()	Hero function that takes in a Space location and assigns it to the hero's current location	Function should correctly assign the current location so hero's location is up to date
checkSpecialItems()	Checks if all 4 special items have been collected	Correctly iterates through first 4 elements of the array and returns true if the items are all there
Memory/Segmentation	Check for any memory leaks and segmentation errors using valgrind, by	Valgrind check should return no leaks No segmentation error

	running simulation with different player combos	should be present after each test

Reflection

I actually spent more time on the design of this project than any of the other projects. Even then, there were changes and additions that I realized I needed as I was coding. I chose a theme that was interesting for me, and made sense in terms of the requirements of the assignment. Some of the choices I made were the number and types of derived spaces and the format of the linked Spaces. I didn't use stl containers because the thought of trying to link 3x3 Spaces and one on either side seemed too complicated. I decided to just link them via their pointers and then have a pointer for the current location. I hard-coded in the map which mostly involved setting up the pointers to point to the correct Space locations in each direction. I also decided to use an array for the backpack container, and then add the items in specific locations. The 4 special items were placed in positions 0-3, while the weapons in places 4-9 after checking for nullptr. A few bugs I caught after playing the game several times, such as the incorrect assignment of a pointer in the map which was an easy fix.

Some of the changes I made as I was coding:

- checkItems function was removed from being in Mordor to Quest.
- checkSpecialItems was removed from Items or Mordor classes.
- The number of days to complete the game was initially 30, but that didn't provide any challenge at all, so I reduced it to 20.
- I wasn't subtracting days for any of the random events, but the game was more interesting when some of the random events decremented the remaining number of days.

Coding challenges I had to work on and fix:

- Input validation was accepting a character even though it wasn't a direction that could be travelled
- currentLocation was stuck at 'shire', the function wasn't being called correctly.
- The last problem I encountered was that when the chooseMenu function ran, the output would initially always be "invalid input … ". I finally traced this error back to the buffer not being cleared after the user entered their name. I solved this by using getline for the name instead of just cin.

I enjoyed coding the final, and I do think that I would like to include more interesting functionality. I think it is modular enough that more derived spaces and events could be added. The one thing I would change is to have the user press enter a few times so that a big wall of text doesn't show up right away!

Thanks for your help and feedback!