

Phillip Wellheuser

CS 162-400

6/11/19

Project 5: Text Based Game

Planning:

This project seems very doable; I know that I know all of the skills that I'll need to develop it, and there's a lot of freedom and flexibility (a blessing and a curse) but if I plan well, it should go pretty quickly (I hope). The main concepts I expect to utilize here are inheritance, containers, and pointers along the lines of the linked lists we created in prior assignments.

In this project there will be a lot going on, so I want to try to focus on easy cleanup so I don't get bogged down in sussing out memory leaks. I'm also going to try to go easy on the creative aspects because it's easy to get carried away in those. I'll try to make my excess creative choices things that can easily be cut in case of a time crunch or unforeseen difficulties.

My game will be a text based adventure as a lone researcher on a space research vessel where a catastrophic damage has occurred and you must escape before oxygen runs out. So the player will have to gather all of the required items necessary to repair a damaged escape pod and take them to the escape pod. I don't to manage a bunch of interactions with things, so I'll be making all interactions that are particular to items the player might have in their inventory activate automatically to lighten the workload on myself.

The two main challenges for the player will be finding everything they need before oxygen runs out and getting through doors that have been locked or otherwise obstructed for some reason or another. But I will cut the locked doors feature if I need to. I'm also going to add an additional mechanic of getting more oxygen in spaces that are sealed properly and losing oxygen in spaces that are not sealed.

Requirements:

1. Base class Space with at least 1 pure virtual function which will make it an abstract class and 4 pointers to Space objects top, right, left, and bottom.
 - a. At least 6 spaces (presumably different inherited classes)
2. A theme
 - a. Mine will be escape space ship
3. Keep track of the space the player is in
 - a. Every iteration of the game cycle will indicate what space the player is in and the rest of the game state
 - b. I might make a map
4. A container for carrying "items"
 - a. Has a capacity limit
 - b. I will make a class that holds a vector and has a way of limiting how many items can be added to it
5. Items
 - a. Must be able to put them into player inventory
 - b. At least 1 must be integral to victory conditions
 - c. I will have several items for repairing the escape pod and escaping
6. "Time limit"
 - a. limits the amount of time/steps/turn the user can take before losing the game

- b. I will use running out of oxygen for this
- 7. Interactions
 - a. The player must do more than just pick up items throughout the game
 - b. In my game there will be opportunities to pick up items as well as examine them and examine the spaces in which the player is currently residing
 - c. Additionally, there will be broken parts of the ship which can be repaired by taking specific items to them, changing the state of the game
- 8. Declare goal of the game at start
 - a. I will have a list of notes/directions at the start of the game\
- 9. No free form input
 - a. All input will be done by choosing number options
 - b. Must provide a menu option for each scenario of the game
- 10. Optional Map
 - a. I might do a map...

Design:

- 1. Main
 - a. Will start the game and give a quit option when complete
- 2. Game class
 - a. Runs the outermost layer of the game, tests for victory conditions
 - b. Shows directions and map
- 3. World class
 - a. Sets up and manages a group of Spaces in which the player will play
 - b. Manages surface level interactions with these objects
- 4. Space class
 - a. Abstract class which is the base class for all spaces in the game
 - b. Has an Inventory which may or may not have Items in it
- 5. Inventory class
 - a. Contains a vector of items which can be added and removed
 - b. Has a variable size limit to how many items it can contain
 - c. Functions for adding and removing items
 - d. Functions for searching to see if items are present in it
- 6. Item class
 - a. Has a description
 - b. Has a name to
 - c. Has an enum to tag what type of item it is
 - d. Rather than creating subclasses the enum list of tags will be used in the constructor to determine what kind of item it should be and then strings for name and description will be defined from a list

Testing:

I'm sure there will be issues with my pointers between spaces at some point, and I always end up cleaning up memory leaks, so I'll keep an eye out for those in particular

1. Do all doors in all spaces connect to somewhere
 - a. All doors, when a menu option is selected should move the player to the new space indicated by the menu when selected.
 - b. There were some issues with connecting to Outer Space due to an error in returning pointers to the space which caused the player never to be able to go into it again after returning to the ship the first time. Repaired this.
 - c. Results: as expected
2. Do all doors in all spaces have proper names in menu
 - a. All doors in each space should display on the menu, have options to use them, and reflect accurately the space which they will move the player to.
 - b. Test by going to each space and testing each door
 - c. Results: working
3. Do all spaces have interactable items in their inventory if they should?
 - a. Some spaces, such as the crew quarters should have items populated into them by default. Can these items be moved and seen in menu?
 - b. Test by going to each room with items by default, and moving their items into player inventory
 - c. Results: working
4. Do spaces with items added late show the items and can items be moved?
 - a. Some spaces, such as the cargo bay have items added to them late after certain tasks are accomplished. Can these items be moved and seen in menu?
 - b. Test by accomplishing the necessary tasks and entering the rooms then moving the items to inventory
 - c. Results: working
5. Does victory condition work?
 - a. Victory is accomplished by reaching the escape pods with all of the required items. Does doing so trigger victory?
 - b. Acquire all required items and go to escape pods
 - c. Results: working
6. Does lose condition work?
 - a. The lose condition is running out of oxygen does this trigger losing?
 - b. Run around the ship, without picking up items until oxygen depletes
 - c. Results: working
7. Do steps and step status update accurately/consistently?
 - a. Steps should deplete with checking inventory and with moving to a new space, and they should do so at different rates depending on whether the player is in Outer Space, an unsealed room or a sealed room with life support functioning
 - b. Test by going through each space with life support turned on and observing step counter
 - c. Results: working
8. Does Oxygen Tank get spent when running out of oxygen with it in inventory?
 - a. When reaching 0 steps, if the player is carrying an extra oxygen tank, then that tank should be removed and 4 steps should be added to the step count

- b. Test by adding an oxygen tank to inventory then running out of steps
 - c. Results: working
- 9. Do all spaces with special interactions update items, strings, and booleans properly upon entering?
 - a. Spaces that have circumstantial string changes and items added to their inventory which are controlled by boolean values. Do these update appropriately?
 - b. Test by going to each room without fulfilling its circumstances and then returning to with those circumstances fulfilled observing the strings and times present.
 - c. Results: working
- 10. Do all items in inventory delete properly? Do all Spaces in World delete properly? Do Bridge and Engines delete properly? Does Outer Space delete properly? Does World delete properly?
 - a. All of these objects are tracked using pointers and should be deleted to deallocate memory upon completing the game or losing.
 - b. Test by completing the game and losing the game while using valgrind
 - c. Results: working
- 11. Are there any memory leaks?
 - a. Test by playing the game through valgrind
 - b. Results: working
- 12. Does game reset properly when playing again?
 - a. Play the game again after completing it to see if it still works correctly
 - b. Results: working

Reflection:

I did not plan this project as well as I would have liked to. I ended up having to take out my features for doors being locked on the ship because, though I had them mostly working, my program was not well suited to implementing them and it became too much of a hassle to make it work. If I'd spent more time planning early on, I may have been able to avoid this or at least foresee the trouble it might cause.

I also don't think I utilized my pure virtual functions to their fullest and that my constructors for my derived classes could have been better organized. I wish I had managed Outer Space so that when the player was in Outer Space it behaved more similarly to the other derived space classes as well.

I also ended up a little more rushed on this project because my CS 271 final assignment was having some peculiar errors, so both design and implementation suffered here I think.

What went well was that I was able to clean up almost all of my memory allocation on the first try. There was one issue where I was trying to delete unallocated memory, but I fixed that by making all of my destructors check to see if their target was already a null pointer before running. It's not the most elegant solution but it works.

As far as I can tell, it fulfills all of the requirements of the assignment and I'm pretty confident turning it in.