

Requirements

- Text based game where player moves through series of rooms or spaces - [haunted cruise ship with different rooms](#)
- Player needs to gather items to achieve a purpose - [journal pages](#)
- Each space/room must be a class with at least 4 pointer variables that link to other spaces. - [yes, each has 4 pointers though some are set to NULL. The pool uses all 4 pointers.](#)
- Must have at least 5 spaces of 3 types - [9 spaces](#)
- Must have an abstract space class with a special pure virtual function - [yes Room has virtual get msg and virtual special](#)
- Each type of space will have a special action - [yes, controls interactivity in room](#)
- Special action includes: control the engine, turn on lights, open doors, etc - [locks doors, prints messages, increments timer, casino game](#)
- Must have at least 3 derived classes for different spaces - [9 derived classes](#)
- Player must have a goal, player must discover a solution - [player must collect 4 journal pages and then play a casino game to reveal final scene](#)
- Must keep track of which space the player is in. - [player is a room pointer](#)
- Player must have a container for items. At least some of these items must be required for the solution. Container should have a limit. - [empty journal](#)
- Structure of the spaces must change. At least one space must be added, and at least one space must be removed. - [cabin is removed, final scene is added](#)
- No free form input to solve puzzles. - [only integer inputs to select a menu item](#)
- Player must interact with the structure beyond just collecting items. - [plays casino games, gets special messages, reveal final scene](#)

Design

Theme

Haunted Cruise Ship with a Big Plot Twist at the end.

You wake up in the cabin of a cruise ship, but you don't know who you are or how you got there. All you have in your possession is a blank journal with torn out pages. In the cabin you find a torn out page. It tells you that you are on your honeymoon. You set out to find your fiancé and ask him what has happened. You leave the cabin (the door locks behind you). The ship seems empty and eerily quiet. Not all the lights are on, and the ones that are, flicker erratically. There is a map on the wall and a fire axe in a glass box labeled "for emergency use only". [Do you want to take the axe? (Y/N) Hint: you should probably take the axe.] You examine the map. Most of the map has been defaced with white paint, but you can see that the hallway you are in ends at a large indoor pool at the center of the ship. [If you try to go back in your cabin, the door is locked.]

You walk down to a large indoor pool at the center of the ship. Ahead, you get a glimpse of the ship's Captain with a scraggly unkempt beard and greasy hair. He laughs in an unsettling manner

and walks through the sliding glass doors. You follow him. When you enter the pool, the doors slide shut behind you and will not re-open. You are really glad you have an axe, now, because something definitely is not right. (You did grab that axe, right?) You see a woman and a small child at the far end of the pool. The woman appears angry with the child, and she is berating it over some misbehavior, but you can't quite make it out. You prop the axe behind a lounge chair, out of view of the woman, and call out, "Hello! Do you know where everyone is?" The woman appears startled to see you. She drags the child out of the pool when she sees you approach. She drags the child down another hall and into a cabin and locks the door, leaving you alone in the pool room.

You notice many corridors from the pool to other parts of the ship. The pool appears to be a hub in the middle of the ship. You notice another journal page sitting on a small table next to the lounge chair. It reveals that you are a software engineer who is the technical co-founder of a new start up. It talks about how excited you are about your tech company. The earnings over the last few months have been phenomenal. You cannot believe that everything would happen so fast. You never thought you would be this rich. It still doesn't seem real. You decide to elope with your fiancé and sneak in a quick honeymoon before everything really gets crazy.

You examine the map on the wall and wonder which way to go. The door you came through is now also locked. However, there are several places to explore. From the pool, you can go to the
1) All-you-can-eat buffet. 2) The exercise room 3) The ship's theater 4) the cocktail lounge.

Each room will contain a spooky creature or ghost that the player has to battle. These will be implemented similarly to the Creatures in assignment 3. If you have the axe your attack power will be increased dramatically. If you fail to defeat a creature then you will be bounced back to the pool area. If you defeat the creature then a door to the ship's casino will be revealed. In the casino is a one-eyed croupier. He offers you a seat at the black jack table. You play a hand of blackjack [I think I can implement Black Jack pretty easily, but if I can't, then Black Jack can be replaced by some type of dice game]. If you win the casino game, you get a page in your journal and learn more of the story, and move back to the pool area. If you lose, you get a message that the ship is taking on water and sinking and get moved back to the pool area. Rooms in which you have defeated the creature will be locked. If the creature has not been defeated, then the room remains unlocked until the creature is defeated.

Periodically, the captain makes a crazy announcement over the ship's public address system, taunting you. Occasionally, you see the angry mom slapping and berating her child. At one point, the angry mom seems to be drowning her child in the pool when you re-enter the pool room. She runs away when you try to stop her. If you lose the casino game X number of times (maybe 10?) then the game is over, the ship sinks. Ship sinking messages, bad mom vignettes, crazy captain announcements, and journal pages will be stored in a queue.

Once the player collects all the journal pages (4 pages) then her full back story has been revealed. A new door from the casino unlocks leading to the observation deck on the back of the ship over looking the ocean. The final scene unfolds and reveals the Big Plot Twist that ties together all the crazy elements of the game. (I promise it all ties together! This is not like Lost, I swear!) The player gets one last chance to make a choice that will either win or lose the game.

Note: somehow I have to work an elevator full of blood into this!

Big Plot Twist/final scene

The last journal page reveals that your fiancé tried to drown you. You are currently in a hospital bed in a coma, and you must wake up so you can tell everybody that it wasn't an accident, and your fiancé is a murderer. You run out to the observation deck just in time to see the bad mom throw her child off the ship. You have two choices, jump in after the child or remain on the ship. If you remain on the ship, the captain comes out with an axe and kills you. You lose. You wake up just as your killer fiancé puts a pillow over your face. If you jump in after the child, you wake up when you hit the ocean.

You wake up in a hospital bed. The crazy captain, bad mom, and 1 eyed croupier are there. The crazy captain is actually your fiancé. The bad mom is his narcissistic mother (your MIL), and the 1 eyed croupier is your doctor. It was your honeymoon. You had a drowning 'accident' aboard your yacht. Your fiancé tried to kill you by drowning you in the hot tub, but botched it, and you survived and were found by the maid floating face down, but in a coma. Now you are awake and you reveal the story to your 1 eyed doctor.

Rooms/Creatures

All you can eat buffet - Blue Men

The all you can eat buffet is filled with rats (Blue Men) that attack you when you enter.

Exercise room - Baba Yaga

Deranged personal trainer attacks you by throwing barbels. She thrives on your pain!

Lounge - Medusa

You head towards the lounge. Something invisible is swishing around you. You swing wildly for the invisible entity. Medusa effect — if you get a glimpse of the apparition, you are frozen with fear.

Theater - Harry Potter

Evil magician hurls unforgivable curses at you.

String queues

Journal

Lost journal pages — finding lost journal pages unlocks final scene

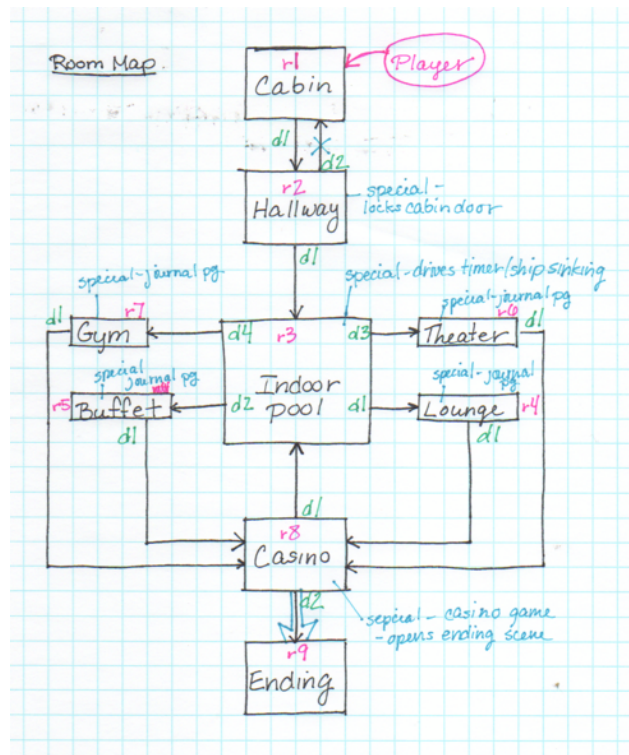
Sinking messages/Captain announcements — acts as game timer

Game Objectives

1. Collect 4 missing journal pages from the theater, lounge, buffet, and gym
2. Explore the casino and play a chance based game
3. Enter the spooky elevator that appears in casino
4. Accomplish all objectives before the ship sinks (triggered by entering or exploring the indoor pool)

Map of Rooms

Notes: Rooms should be designed so that choosing direction 1 will always move the player to a new room. Too many dead ends would be too frustrating. Once entering the pool, player should not be able to turn back. Player should encounter Casino and Pool frequently to progress through the game objectives in a reasonable fashion.



Implementation - Stage 1

1. Create abstract room class and add derived rooms. Each room will have a message describing the room. Each room has pointers to other rooms, and a special function
2. Create a room layout. Maneuver room pointer / player around the rooms using hardcoded commands.
3. Create a do/while loop #1 that allows player to interactively maneuver through rooms. In main, choices: explore room (special()), move, read journal, quit.
4. Do/while loop #2 lets player pick a direction and returns a pointer to that direction.
5. Create a player class to hold room pointer, journal, health/life stats.
6. Implement journal with DynQ class.
7. Add/remove rooms.
8. Implement casino with dice game. When player wins game new room is created.
9. Sinking queue. This is the "timer". Once it is empty, the game is over.
10. Improve each room's special function.
11. Implement creature battles.
12. Refactor Casino to have a Black Jack game.
13. Improve journal entries and print out messages for a better story

Testing

Unit testing

DynQ (Journal, Sinking Messages)
See assignment 4.

Room Class

Hardcoded derived rooms (Cabin and Hall) to test virtual Room functions

Test Case	Input Value	Driver Function	Expected Output	Observed Output
Get cabin msg	r1->getMsg();	getMsg()	"This is the cabin"	✓
Get hall msg	r2->getMsg();	getMsg()	"This is the hall"	✓
Room interactivity	r1->special()	special()	"This is hall special"	✓
Traversing rooms with player pointer	r1->setD1(r2); player = r1; player->getD1();	getMsg()	"This is the hall"	✓

Casino Game

Test Case	Input Value	Driver Function	Expected Output	Observed Output
Play game	1	Casino::special()	a random number from 1-6	✓
win game	—	Casino::special()	Win message	✓
lose game	—	Casino::special()	Allow player to play again or quit	✓
Quit game	2	Casino::special()	Immediate exits game without asking for additional inputs	✓

Integration testing

Test Case	
Movement	✓
Room message displays correctly when each room is entered.	✓
Rooms are interconnected as depicted in map.	✓
Trying to go a direction that doesn't exist results in an error	✓
Messages are clear and informative	✓

Interactivity/Special function	✓
Correct special message plays for every room	✓
Special - Hallway - locks cabin door, player cannot go back into cabin	✓
Special - Pool - plays a sinking or story message.	✓
Special - lounge - reveals 1 journal page	✓
Special - buffet - reveals 1 journal page	✓
Special - theater - reveals 1 journal page	✓
Special - gym - reveals 1 journal page	✓
Only 1 journal page discoverable per room	✓
Special - Casino - dice game is played until player wins or quits	✓
Ending Sequence	✓
If 4 journal pages have been found and casino game is played, new room is added (an 'elevator' in casino direction 2)	✓
Entering elevator displays final scenes and ends game with 'win' message.	✓
Entering/exploring the pool more than 10 times ends game with a 'lose' message.	✓
Quit option exits game immediately.	✓
Messages are informative enough to help player maneuver structure	✓
Story flows well and makes sense (even if it is not a literary masterpiece)	✓
Input validation: only valid entries are accepted. User is prompted to re-enter menu selection if the choice was invalid	✓

Changes/Bugs

1. Player is a room pointer in main();
2. Changed the move and explore functions to make game play more intuitive.
3. Change some room messages when the room is re-entered.
4. Added the ability to read the journal with readJournal() function. *-edit: removed, bugs*
5. Decided not to implement player class
6. Experimented with creating a cruise ship class to hold all the rooms, but ran into problems. Decided not to use cruise ship class.
7. Dynamic vs Static rooms. Most of the rooms of the ship were declared statically at the beginning of implementation. The static rooms can have pointers to/from them removed, but they are not completely removed until the program terminates. Therefore, some rooms will be dynamic. There are the rooms that will be added/deleted during runtime using the special function in adjacent room. The hall special()/explore removes the cabin. The casino adds the

final scene if all journal pages have been collected. Dynamic binding is somewhat slower than static binding so having the majority of the rooms statically declared would improve the playability of the game if the game were expanded to include large numbers of rooms.

8. The cabin was changed to a dynamic room so it could be deleted when the player explored the hall.
9. Found a bug: can only read the journal 1 time or else it crashes. Journals should probably not have been queues; they should have been implemented as arrays or even possibly maps, or something else that allows you to iterate through them easily without removing contents. For now, disable readJournal. Print out journal entries as pages are found instead.

Reflections

Container choices — I probably did not pick the best container type for the journal. An array or a map would have been a better choice. After starting the program, I realized that queues are not great for iterating through and revealing the content without erasing it. I ran into a serious bug with my readJournal function and could not fix it. In the interest of time, I decided to do away with it. An array of strings would probably have been easier to work with.

Design — One challenge that made this assignment difficult was that bugs and extra features kept expanding the scope of this project. I was over ambitious and over designed this project. I should have kept it simpler to start with. I think it would have worked better to build a skeletal program with linked rooms and the game loop, then written the story to fit the format instead of writing the story first and trying to get my limited programming skills to fit the story. Ultimately, I had to reduce the scope of my project and focus on getting something that met requirements and could be turned in. Also, the story turned out really dark. I didn't mean to make anything disturbing. In retrospect, I wish I had tried to write something funny instead.

Re-usable code — I should have made the rooms more modular and re-usable. I probably only needed 4 different types of rooms to implement my design and then just given them unique names. That would have sped up development of the game considerably.

Encapsulation — Most of the game logic is in main, and the whole thing feels disorganized. On future versions, I would improve code style by encapsulating more of the game logic in the methods of a separate class that runs the game. I attempted that with the cruise ship class, but was not able to implement it before the due date.

API — This assignment really demonstrated the importance of a standardized interface. All classes have the same interface so that they can be accessed in the same way by the primary game loop.

Overall, I think this is not quite a complete game, but it is a good skeleton of a game. I think improving the existing story, or even coming up with a whole new story would improve the playability significantly. There are numerous places where the code could be improved and more features added to make a more interesting experience for the player.

Future improvements

- Implement journal as array and have a readJournal function that iterates through the array.
- Have more abstract room types with fewer classes so that some of them can be reused to make other parts of the ship. Instead of a separate class for each room, have more generic types that I can give a unique identifier or name to.
- I probably would have designed the data structure first and then come up with a story line. That seems like it would have been easier to limit scope of this project.
- Implement creature battles.
- Implement a player class to track health and facilitate power ups.
- In casino, replace dice game with black jack.
- Make winning the casino game a requirement for opening the final scene.
- I would have liked to further polish the story. I feel like it has some potential but ultimately it really falls flat in the current implementation.
- If I had more time I would meet with some TAs again for more advice on how to clean up my code.

Acknowledgements

Thank you to Niza Volaire for assistance with this program.

Resources

Colossal Cave adventure: https://en.wikipedia.org/wiki/Colossal_Cave_Adventure

Interactive fiction engines — tads, hugo, inform
<http://www.brasslantern.org/writers/howto/chooselang.html>

Beginning C++ Through Game Programming by Michael Dawson

Complicated Text based game example:
<http://www.penguinprogrammer.co.uk/rpg-tutorial/introduction/>

Simple text based game:
<https://levelskip.com/classic/Make-a-Text-Based-Game>

Game loops
<https://www.quora.com/What-type-of-loop-do-I-need-to-make-a-text-based-game-in-C++>

<http://www.cplusplus.com/forum/beginner/106563/>

<http://www.cplusplus.com/forum/beginner/106563/>

<https://github.com/Hopson97/Hopson-Land/blob/master/main.cpp>

<http://stackoverflow.com/questions/13505617/checking-for-empty-rooms-in-an-adventure-game>

Traversing Linked lists
<http://www.cprogramming.com/tutorial/lesson15.html>

Are text based games harder than graphics based game? Some internet guy says yes.

<http://www.cplusplus.com/articles/G13hAqkS/>