

Final Project

Text-based Game

Goals

- Design a program to satisfy provided requirements
- Use good OOP style, including inheritance, polymorphism, and pointers

Assignment

This is your final project and you have freedom to decide what you want to do. At the end of the term, you will surprisingly find how capable you are to develop a real computer game for your friends to play with! You will design and implement a text-based game where the player moves through a series of rooms or spaces. They will need to gather items to achieve some purpose. You need to meet the specific requirements in your program, and all the other implementation details are left for you to decide.

Specific requirements

You will create a series of rooms or spaces for the player to move through. (By default this is a one-player game. If you want to add more players, you can extend your game later.) Each space will be a class with at least **four pointer variables** (or arrays of pointers) that link to other spaces (Right, Left, Top, Bottom, etc.). Even if your structure is linear, such as a train, you will still have at least four pointer variables in the class. Be creative and try to use them all! Note that any unused pointers will point to NULL. (The requirement for four pointers in each space is to simplify coding. If you need more pointers for your theme, feel free to add more.)

The space will also have appropriate data members. You must have **at least 6 spaces of at least 3 different types**. You will have a space **abstract class** that will have special pure virtual functions. Each type of space will have a special action. It could be the controls in the engine, or to control access to one of the doors (pointers), or it could simply turn on the lights. So you will have at least **3 derived classes** for your spaces. To continue with the train theme, you could have a passenger car, or baggage car. Or one space could have controls such as the engine.

You must have **a goal for the player in your theme**. Maybe it is to solve a crime (like Clue?). Maybe the aliens have invaded your spaceship and normal weapons do not work but broccoli causes them to leave! You may have a random goal (so one time broccoli works, next time it is a pillow). Based on your theme, the player must discover the solution to win the game.

You must have some way to **keep track of which space the player is in**. The player will have a **container** (backpack, knitting bag, or notebook) to carry “items”. The container must have some limit (weight, size, etc.). One or more of these **items will be required as part of the solution**, such as a “key” to open the locked door.

For simplicity, your game **should not require free-form input**. It is frustrating if you did not spell something correctly. You should have a **time limit** to urge the player on as well. This does not mean a literal clock, just some way to prevent the game from going on indefinitely. (Make sure you give enough time to allow the game to perform through testing.) The player must **interact with parts of the space structure**, and not just simply collecting things. This can be throwing something at the monster, operating a light switch (or other control), opening doors, or singing to get the baby back to sleep. That means, when the player is moving around, he would not only get information from the spaces, but also change the status of the spaces.

To make it easy for your TA to grade the project, you must provide a **menu option** similar as what you did for previous assignments. Before the beginning of the game, **reveal the goal for the player**. If the grader does not know the goal, they cannot tell if the program is working correctly. This information does not need to be elaborate. Print something like “the broccoli causes the aliens to leave”, or “the murder was done by Mr. Duck, club car, fishing pole.”

As long as you meet those requirements, all the other details are left for you to decide. You must develop a theme for your game and try to make it interesting. Also, the graders will vote for the **best 10 themes** in the class and those students will get **10% of extra credit** for the final project.

What you need to submit:

- All the source and header files
- Your makefile
- Your reflection document
- Pack them all in a zip file and submit it to TEACH.

Suggestions:

Do not spend too much time designing an interesting and fun-playing theme. We are not testing your game designing abilities, but your programming skills. Focus more on your code and make it solid. Try to finish the basic functions that meet all the requirements first, and if you have some extra time, add more functions and make it interesting. Also, always think and design your program and test plans before you write the code.

For the final project, the grading time is very limited. So there is no more time for regarding or email exchange from you and TAs. Make sure you test your program on the flip and there is no segmentation fault. (If you use other IDEs to develop the program, you must do this step!) The final project does not allow late submission. So again, finish a simple version first and extend it later if time is allowed.

GRADING

- Programming style and documentation: 10%
- Create basic game structure 45%:
 - ✓ Create at least 6 spaces of at least 3 different types (the abstract and derived classes): 20%
 - ✓ Correctly use at least 4 pointers for each space to link them together: 10%
 - ✓ Implement a goal for the player to achieve (sequence of actions to win and exit): 10%
 - ✓ Keep track of player status: 5%
- Meet the advanced requirement 15%:
 - ✓ Properly implement the time limit: 5%
 - ✓ Properly implement the interaction of player and space structures: 10%
- Create a menu function to run the game: 10%
- Reflection document 10%: include the design description, test plan and results, comments about how you solved the problems
- Other basic requirements 10%: program runs smoothly as intended on flip, no segmentation fault, no memory leak, all files are put into a zip file.
- Best theme +10%: the graders will have the option to give extra credit for the most creative, interesting and well-designed game in the class.