

Matt Dienhart  
ONID: 933341964  
Final Project Reflection

### Part 1: Program Design

#### **Starship Bloopers**

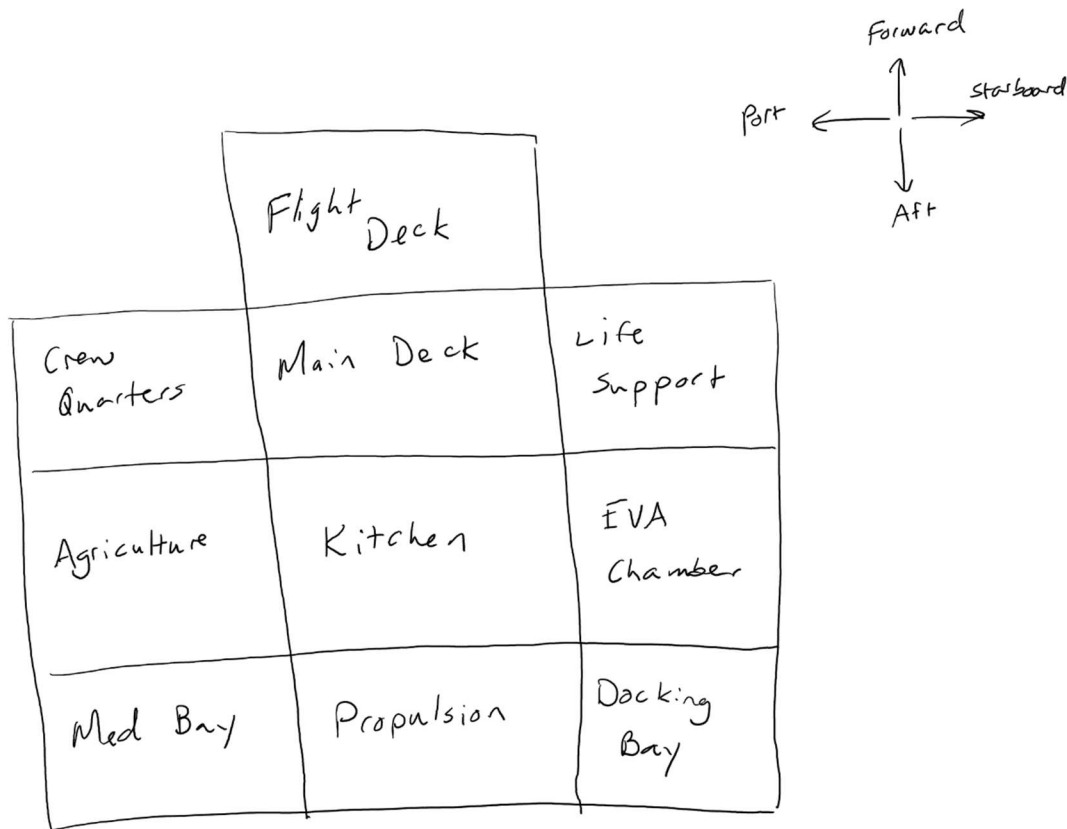
##### **Overview:**

This is a text-adventure style game where you are an officer aboard the military spaceship Daedalus. Your commanding officer and most of the crew has left the ship to attend a fancy banquet on a nearby space station, leaving you and a few of your peers in charge. The game begins when you awaken after a long night of alcohol-fueled mischief around the empty ship. You have to navigate through the different modules of the ship, cleaning up the mess you and your crewmates made last night, before Commander Fitzpatrick returns and punishes you for your bad behavior.

There are 10 spaces in the ship:

- Flight Deck
- Main Deck
- Crew Quarters
- Agriculture
- Med Bay
- Kitchen
- Propulsion
- Life Support
- EVA Chamber
- Docking Bay

You begin in the crew quarters. The map of the ship is shown below. You can ask your character to bring up a map of the ship on their standard-issue data pad at any time, and it will show your location marked on the map.



You are hung over and can't think very well. You need to get to the Med Bay and take some medicine to fix this. If you don't take any medicine, it will take you twice as long to move from space to space or perform any actions.

There are 5 problems in the ship that need addressing:

- There is a graffiti drawing on the outer hull of the ship. You need to wash it off, or else your commander will see it on her way back. This requires obtaining the **EVA Suit** from the EVA Chamber, and interacting with the Docking Bay.
- The ship has drifted into an asteroid field, which is kind of dangerous. You need to get the **Navigation Data** from Propulsion and then interact with the Flight Deck to escape the asteroid field.
- There is a fire in one of the trash cans in Life Support. You need to get the **Fire Extinguisher** from the Kitchen and interact with Life Support to put out the fire.
- One of your crew members stole Commander Fitzpatrick's prized Laurence Fishburne bobblehead from the flight deck, and it's locked in your locker in the Crew Quarters. You need to retrieve the **Laurence Fishburne Bobblehead** from the Crew Quarters and interact with the Flight Deck to put it back.

- The EVA Chamber and some of the EVA suits are covered in spilled beer. You need to get a **Spill Kit** from Agriculture and interact with the EVA Chamber to clean the room and the suits. This is how you get the **EVA Suit**.
- The Main Deck is full of empty beer bottles. You need to collect the **Beer Bottles** and interact with the Docking Bay to flush them out into space.

The game will present you with a set of numbered options every turn:

1. Move
2. Look Around
3. Check Inventory
4. Check Directory
5. Check Watch
6. Interact (available only in certain cases after looking around)

"Move" allows you to move from one space to an adjacent space by specifying a direction. Moving uses 1 minute of time.

"Look around" shows details of the room you are in, and allows you to see what may happen when you choose "interact". For example, "looking around" in the Main Deck will reveal that there are lots of empty beer bottles, and then option #6 appears as "pick up beer bottles". Looking around uses 1 minute of time.

"Check Inventory" shows the contents of the player's inventory array. It just prints the names of everything currently in inventory. It does not cost any time.

"Check Directory" prints the player's current location in a map of the ship. It does not cost any time.

"Check Watch" prints the current time. The game begins at 2pm. When it is 4pm, the game will end.

"Interact" performs a room-specific action. If the room has no room-specific action that can be performed, or if the room has not been inspected yet, then "Interact" will not be selectable. If the room has been inspected, and there is some action that can be taken, this option will appear with a descriptive label (such as "gather beer bottles"). The amount of time it takes to interact depends on the action being performed.

#### Space class pseudocode:

protected:

forward pointer  
aft pointer  
port pointer  
starboard pointer  
name of the space (flight deck, crew module, etc.)  
bool to track whether this space has been inspected yet  
int to track how many actions are left to do in this space  
inspection reminder message (reminder of what was going on in this room)  
interaction label

public:

setters  
getters

virtual function: inspect ("look around")  
virtual function: interact (changes with context)

#### **Player class pseudocode:**

private:

character name  
vector holding player's inventory (string objects)  
player's date of birth (string)  
bool tracking whether player has taken medicine yet  
int timer  
int points earned (each mess cleaned earns 1 point, there are 6 possible points)

public:

setters

- need special functions to add and remove items from inventory

getters

check inventory  
check the ship's directory (prints the map shown above)  
check the time  
search the inventory for a target object

#### **Game class pseudocode:**

private:

pointer to track the player's current location  
pointer to track the crew quarters location (static reference, similar to the HEAD pointer in a linked list)  
pointer to track where the player is going next  
pointer to track the player object

public:

play game function  
valid input function  
move player

constructor:

- the constructor needs to build the linked structure of the ship using dynamic memory allocation
- the ship is always the same layout, but each space is unique, so it may not be possible to do this in a loop
- constructor also needs to create the player object through dynamic allocation

destructor:

- the destructor needs to do the reverse of what the constructor does, freeing dynamic memory one space at a time

- also need to delete the player object

**Game flow:**

Each turn is performed as follows:

- Display the name of the current room, and if the room has previously been inspected and/or interacted with, show the inspection reminder.
- Prompt the user to select an option.
- Process the user's option, including the passage of time, and awarding of points if the player successfully cleaned up one of the messes.
- At the end of the turn, check to see if time is up or all 6 messes have been cleaned - if so, display the won/lost screen.

### Part 3: Test Plan

Test Case	Input Value	Driver Functions	Expected Outcome	Actual Outcome
string input	"foo"	main() play or quit selection	Error, repeat prompt	Error, repeat prompt
invalid integer input	-3		Error, repeat prompt	Error, repeat prompt
double input	1.5		decimal is truncated, program calls Game.playGame()	decimal is truncated, program calls Game.playGame()
too high input	3		Error, repeat prompt	Error, repeat prompt
too low input	0		Error, repeat prompt	Error, repeat prompt
highest valid input	2		program calls Game.playGame()	program calls Game.playGame()
lowest valid input	1		program terminates	program terminates
string input	"foo"	after game completes, main() play again or quit selection	Error, repeat prompt	Error, repeat prompt
invalid integer input	-3		Error, repeat prompt	Error, repeat prompt
double input	1.5		decimal is truncated, program calls Game.playGame(), no data remains from previous game	decimal is truncated, program calls Game.playGame(), no data remains from previous game
too high input	3		Error, repeat prompt	Error, repeat prompt
too low input	0		Error, repeat prompt	Error, repeat prompt
lowest valid input	1		program calls Game.playGame(), no data remains from previous game	program calls Game.playGame(), no data remains from previous game
highest valid input	2		program terminates	program terminates
string input	"foo"	Game.playGame() menu selection (room not inspected yet)	Error, repeat prompt	Error, repeat prompt
invalid integer input	-2		Error, repeat prompt	Error, repeat prompt
double input	1.5		decimal is truncated, program proceeds with "move"	decimal is truncated, program proceeds with "move"
too low input	0		Error, repeat prompt	Error, repeat prompt

lowest valid integer input	1	Game.playGame() menu selection (room has been inspected and there are actions remaining)	program advances with "move"	program advances with "move"
highest valid input	5		program advances with "check time"	program advances with "check time"
too high input	6		Error, repeat prompt	Error, repeat prompt
string input	"foo"		Error, repeat prompt	Error, repeat prompt
invalid integer input	-3		Error, repeat prompt	Error, repeat prompt
double input	1.5		decimal is truncated, program proceeds with "move"	decimal is truncated, program proceeds with "move"
too high input	7		Error, repeat prompt	Error, repeat prompt
too low input	0		Error, repeat prompt	Error, repeat prompt
lowest valid input	1		program advances with "move"	program advances with "move"
highest valid input	6	Game.movePlayer() option display changes based on which directions are available	program advances with "interact"	program advances with "interact"
In the Flight Deck, can only move Aft	currentLoc = Flight Deck		only one movement direction displayed: "1. Aft"	only one movement direction displayed: "1. Aft"
In the Main Deck, can move in any direction	currentLoc = Main Deck		four movement directions displayed: "1. Forward, 2. Aft, 3. Port, 4. Starboard"	four movement directions displayed: "1. Forward, 2. Aft, 3. Port, 4. Starboard"
In the Crew Quarters, can only move Aft and Starboard	currentLoc = Crew Quarters	Agriculture.inspect()	two movement directions displayed: "1. Aft, 2. Starboard"	two movement directions displayed: "1. Aft, 2. Starboard"
Not inspected yet	inspected = false		interact is not an option, "you have not looked in this room yet"	interact is not an option, "you have not looked in this room yet"
Inspected but not interacted with	actionsRemaining = 1		interact is an option, reminder says there is a spill kit available	interact is an option, reminder says there is a spill kit available
Inspected and interacted with	actionsRemaining = 0	Agriculture.interact()	interact is not an option, "there is nothing more to do here"	interact is not an option, "there is nothing more to do here"
Not yet interacted with	actionsremaining = 1		Spill Kit is added to inventory	Spill Kit is added to inventory
already interacted with	actionsremaining = 0		"there is nothing more to do here"	"there is nothing more to do here"
Not inspected yet	inspected = false	MedBay.inspect()	interact is not an option, "you have	interact is not an option, "you have

			not looked in this room yet"	not looked in this room yet"
Inspected but not interacted with	actionsRemaining = 1		interact is an option, reminder says there is medicine available	interact is an option, reminder says there is medicine available
Inspected and interacted with	actionsRemaining = 0		interact is not an option, "there is nothing more to do here"	interact is not an option, "there is nothing more to do here"
Not yet interacted with	actionsremaining = 1	MedBay.interact()	Player medicated variable = true	Player medicated variable = true
already interacted with	actionsremaining = 0		"there is nothing more to do here"	"there is nothing more to do here"
Not inspected yet	inspected = false	CrewQuarters.inspect()	interact is not an option, "you have not looked in this room yet"	interact is not an option, "you have not looked in this room yet"
Inspected but not interacted with	actionsRemaining = 1		interact is an option, reminder says the bobblehead is available	interact is an option, reminder says the bobblehead is available
Inspected and interacted with	actionsRemaining = 0		interact is not an option, "there is nothing more to do here"	interact is not an option, "there is nothing more to do here"
Not yet interacted with	actionsremaining = 1	CrewQuarters.interact()	Player is asked to enter passcode, bobblehead added to inventory if passcode is entered correctly	Player is asked to enter passcode, bobblehead added to inventory if passcode is entered correctly
already interacted with	actionsremaining = 0		"there is nothing more to do here"	"there is nothing more to do here"
Not inspected yet	inspected = false	MainDeck.inspect()	interact is not an option, "you have not looked in this room yet"	interact is not an option, "you have not looked in this room yet"
Inspected but not interacted with	actionsRemaining = 1		interact is an option, reminder says there are beer bottles to pick up	interact is an option, reminder says there are beer bottles to pick up
Inspected and interacted with	actionsRemaining = 0		interact is not an option, "there is nothing more to do here"	interact is not an option, "there is nothing more to do here"
Not yet interacted with	actionsremaining = 1	MainDeck.interact()	Beer Bottles are added to player's inventory	Beer Bottles are added to player's inventory
already interacted with	actionsremaining = 0		"there is nothing more to do here"	"there is nothing more to do here"
Not inspected yet	inspected = false	Kitchen.inspect()	interact is not an option, "you have	interact is not an option, "you have



			not looked in this room yet"	not looked in this room yet"
Inspected but not interacted with	actionsRemaining = 1		interact is an option, reminder says there is a fire extinguisher to collect	interact is an option, reminder says there is a fire extinguisher to collect
Inspected and interacted with	actionsRemaining = 0		interact is not an option, "there is nothing more to do here"	interact is not an option, "there is nothing more to do here"
Not yet interacted with	actionsremaining = 1	Kitchen.interact()	Fire Extinguisher is added to player's inventory	Fire Extinguisher is added to player's inventory
already interacted with	actionsremaining = 0		"there is nothing more to do here"	"there is nothing more to do here"
Not inspected yet	inspected = false	Propulsion.inspect()	interact is not an option, "you have not looked in this room yet"	interact is not an option, "you have not looked in this room yet"
Inspected but not interacted with	actionsRemaining = 1		interact is an option, reminder says there is navigation data to collect	interact is an option, reminder says there is navigation data to collect
Inspected and interacted with	actionsRemaining = 0		interact is not an option, "there is nothing more to do here"	interact is not an option, "there is nothing more to do here"
Not yet interacted with	actionsremaining = 1	Propulsion.interact()	Navigation Data is added to player's inventory	Navigation Data is added to player's inventory
already interacted with	actionsremaining = 0		"there is nothing more to do here"	"there is nothing more to do here"
Not inspected yet	inspected = false	LifeSupport.inspect()	interact is not an option, "you have not looked in this room yet"	interact is not an option, "you have not looked in this room yet"
Inspected but not interacted with	actionsRemaining = 1		interact is an option, reminder says there is a fire to put out	interact is an option, reminder says there is a fire to put out
Inspected and interacted with	actionsRemaining = 0		interact is not an option, "there is nothing more to do here"	interact is not an option, "there is nothing more to do here"
Not yet interacted with, player has fire extinguisher	actionsremaining = 1, inventory includes Fire Extinguisher	LifeSupport.interact()	Fire Extinguisher removed from player's inventory, player gains 1 point	Fire Extinguisher removed from player's inventory, player gains 1 point
Not yet interacted with, player does not have fire extinguisher	actionsRemaining = 1, inventory does not include Fire Extinguisher		"You do not have what you need to resolve this issue."	"You do not have what you need to resolve this issue."

already interacted with	actionsremaining = 0		"there is nothing more to do here"	"there is nothing more to do here"
Not inspected yet	inspected = false	EVACHamber.inspect()	interact is not an option, "you have not looked in this room yet"	interact is not an option, "you have not looked in this room yet"
Inspected but not interacted with	actionsRemaining = 1		interact is an option, reminder says there is a spill to clean	interact is an option, reminder says there is a spill to clean
Inspected and interacted with	actionsRemaining = 0		interact is not an option, "there is nothing more to do here"	interact is not an option, "there is nothing more to do here"
Not interacted with, player has spill kit	actionsremaining = 1, inventory includes Spill Kit	EVACHamber.interact()	Spill Kit removed from player's inventory, player gains 1 point	Spill Kit removed from player's inventory, player gains 1 point
Not yet interacted with, player does not have spill kit	actionsRemaining = 1, inventory does not include Spill Kit		"You do not have what you need to resolve this issue."	"You do not have what you need to resolve this issue."
already interacted with	actionsremaining = 0		"there is nothing more to do here"	"there is nothing more to do here"
Not inspected yet	inspected = false	DockingBay.inspect()	interact is not an option, "you have not looked in this room yet"	interact is not an option, "you have not looked in this room yet"
Inspected but not interacted with	actionsRemaining > 0		interact is an option, reminder says there are still unresolved problems	interact is an option, reminder says there are still unresolved problems
Inspected and interacted with	actionsRemaining = 0		interact is not an option, "there is nothing more to do here"	interact is not an option, "there is nothing more to do here"
Actions are not finished, player has EVA Suit	actionsremaining > 0, inventory includes EVA Suit	DockingBay.interact()	EVA Suit removed from player's inventory, player gains 1 point	EVA Suit removed from player's inventory, player gains 1 point
Actions are not finished, player has Beer Bottles	actionsRemaining > 0, inventory includes Beer Bottles		Beer Bottles removed from player's inventory, player gains 1 point	Beer Bottles removed from player's inventory, player gains 1 point
Actions are not finished, player does not have beer bottles or EVA suit	actionsRemaining > 0, inventory does not include Beer Bottles or EVA Suit		"You do not have what you need to resolve this issue."	"You do not have what you need to resolve this issue."
already interacted with	actionsremaining = 0		"there is nothing more to do here"	"there is nothing more to do here"
Not inspected yet	inspected = false	FlightDeck.inspect()	interact is not an option, "you have	interact is not an option, "you have

			not looked in this room yet"	not looked in this room yet"
Inspected but not interacted with	actionsRemaining > 0		interact is an option, reminder says there are still unresolved problems	interact is an option, reminder says there are still unresolved problems
Inspected and interacted with	actionsRemaining = 0		interact is not an option, "there is nothing more to do here"	interact is not an option, "there is nothing more to do here"
Actions are not finished, player has the bobblehead	actionsremaining > 0, inventory includes bobblehead	FlightDeck.interact()	Bobblehead removed from player's inventory, player gains 1 point	Bobblehead removed from player's inventory, player gains 1 point
Actions are not finished, player has Navigation Data	actionsRemaining > 0, inventory includes Navigation Data		Navigation Data removed from player's inventory, player gains 1 point	Navigation Data removed from player's inventory, player gains 1 point
Actions are not finished, player does not have bobblehead or navigation data	actionsRemaining > 0, inventory does not include Bobblehead or Navigation Data		"You do not have what you need to resolve this issue."	"You do not have what you need to resolve this issue."
already interacted with	actionsremaining = 0		"there is nothing more to do here"	"there is nothing more to do here"
Player completes an action successfully	N/A	Game.playGame() points tracking	increase player's points by 1	increase player's points by 1
Player fails to complete an action	N/A		do not increase player's points	do not increase player's points
Player wins	timer <=120 and points >= 6	Game.playGame() end of game screen	display win dialogue	display win dialogue
Player runs out of time	timer >= 120 and points < 6		display lose dialogue	display lose dialogue
playGame() continues until timer exceeds 120 minutes or player gets 6 points	timer > 120 or points >=6	Game.playGame() round iteration	program continues until time runs out or player wins	program continues until time runs out or player wins
after several rounds of play with various choices		main() program execution	no errors or memory leaks	no errors or memory leaks

## Part 4: Reflection

My original code design called for the Player class to keep track of the player's movement through the spaces, which mean that there was a `#include "Space.hpp"` in the Player class header. The Space class also needs to interact with the player in order to check the player's inventory and manage the timer, so it has a `#include "Player.hpp"` in its header file. When I initially tried to compile the program using this approach, I kept running into a problem where one of the classes would not be recognized as a valid data type. I eventually found out that this was due to circular references: the C++ compiler has to be able to fully resolve one class definition before moving to the next one. So I had to modify my Player class to no longer depend on `"Space.hpp"` and allow movement to be handled by the Game class instead.

Every Space object in this program represents a room in the spaceship, and since each of them has a unique description and way that it can be interacted with, I found that I needed to create 10 unique subclasses that would each only be used once. This means that the constructor for the Game class has to build the Spaces in the ship layout using sequential statements, rather than a loop (and the same in reverse for the destructor).

Each space keeps track of what has been done and what is remaining to be done, so that the player can arrive at the space, realize they need to go get something on another area of the ship, and come back later to fix the problem. There is a string variable in each space object called `"inpsctionReminder"` that displays a summary of what the status of that room is each time the player enters it, and it will update whenever the player makes progress until everything is done.

Some of the rooms have multiple actions that can be taken, so in those cases, I had to allow for the possibility that the player might only have the necessary inventory items to do a subset of the actions. There is a variable in these Space objects called `"actionsRemaining"` that counts down any time the player makes progress, until it reaches 0. If `actionsRemaining = 0` in any Space, the player will not be allowed to `"interact"` with that Space anymore.

Some of the Spaces require the user to solve a puzzle in order to make progress. I tried to make these fairly simple because I didn't want it to take too long to complete the game (since that would make it harder to troubleshoot).

I started with a time limit of 90 minutes, and since each move, inspect, and interact command takes time, I found that 90 minutes wasn't enough to actually allow someone to finish the game. So I increased the time limit to 120 minutes and reduced the time taken by some of the actions. That gives the player enough time to clean up everything with about 15 minutes to spare, but only if they take the medicine in the Med Bay right at the start of the game.