Jason Hoffman
12/8/2015
Final Project, CS162

Project Description

Note: A map of the "world" is included at the end, along with instructions on how to test/play through the game efficiently.

Planeworld (my temporary) name, is a text based adventure game in which the character moves throughout an airplane looking for their lost iPhone. Along the way, the character will talk with other passengers, interact with the environment, and gather objects that allow progress through the environment. The game will take the form of a puzzle, with clues providing evidence for how to proceed.

Plan for Classes

Space

The assignment requirements call for a Space abstract class from which all locations in the game will inherit. Pointers will be used to connect Space objects. In this game, each Space will have *fwd, back, left* and *right* pointers. Core Space objects will allow movement in four directions, while edges will point to NULL. To interact with the environment, Space will contain three virtual functions (two pure virtual): *talk(), lookAround()* and *manipulate().*

Space

protected member variables:
  Space *fwd
  Space *back
  Space *left
  Space *right
  Character *character
  std::string spaceName

public member functions:
  Space() (constructor)
  void setLocation(Space* f, Space* b, Space* l, Space* r, Character *c, std::string n)
  Space* getFwd()
  Space* getBack()
  Space* getLeft()
  Space* getRight()

```
  Character* getCharacter()
  std::string getSpaceName()
  void setSpaceName(std::string)
  virtual void talk()
  virtual void lookAround()
  virtal void manipulate()
```

Character

The Character object will represent the game's main character. The object will have a *name* string member variable, and a vector of strings to hold objects acquired during gameplay. Member functions will include *moveFwd(), moveBack(), moveLeft()* and *moveRight().* A *currentSpace* member variable will also be included to represent the character's current Space.

Character

protected member variables:
  std::string name
  Space *currentSpace
  std::vector<std::string> objects

public member functions:
  Character()
  void setName(std::string)
  void setCurrentSpace(Space *s)
  Space* getCurrentSpace()
  std::string getName()
  bool objectSearch(std::string o)
  void moveFwd()
  void moveBack()
  void moveLeft()
  void moveRight()
  void addObject(std::string o)

PlaneWorld

To manage the size of main.cpp, a PlaneWorld object will be created in which all spaces will be instantiated, along with Character. Pointers between Space objects will be set up here to create the "world". PlaneWorld will hold the primary pointer to Character object, which will then be used in each Space.

PlaneWorld
private member variables: Character *character;

public member functions:
  PlaneWorld()
  Character* getCharacter()

Game setup

To create the game environment, all Spaces will be instantiated in PlaneWorld and pointers set to adjacent spaces. Each Space object will contain a pointer to PlaneWorld's main Character object. In main.cpp, PlaneWorld will first be instantiated, then its character object stored in a pointer in main.cpp for use in the game.

To further manage the size of main.cpp, a function, *gameLoop()*, will be used in main to run the game. Exiting the game will exit the loop. Most of the dialogue and interaction will be contained in the different Space derived classes.

Planned Spaces

The following is a list of Space objects that will be used to make up PlaneWorld

Back Lav - Rear starting point for the character at the back of the airplane

Back - Rear core Space with forward, back, left and right pointers

Drink Cart - Space pointed to by Back where first interaction occurs

Secret Cargo - Locked space pointed to by Back where game ends

Economy Class - Second core Space pointed to by Back with forward, back, left and right pointers

Crying Baby - Space pointed to by Economy Class where interaction occurs

Drooling Sleeper - Space pointed to by Economy class where interaction occurs

First Class - Core Space pointed to by Economy class with forward, back, left and right pointers

Justin Bieber - Space pointed to by First Class where interaction occurs

Arnold Schwarzenegger - Space Pointed to by First Class where interaction occurs

Forward - Core Space pointed to by First Class with forward, back, left and right pointers

Front Galley - Space pointed to by Forward where interaction occurs

Front Lav - Space pointed to by Forward where interaction occurs

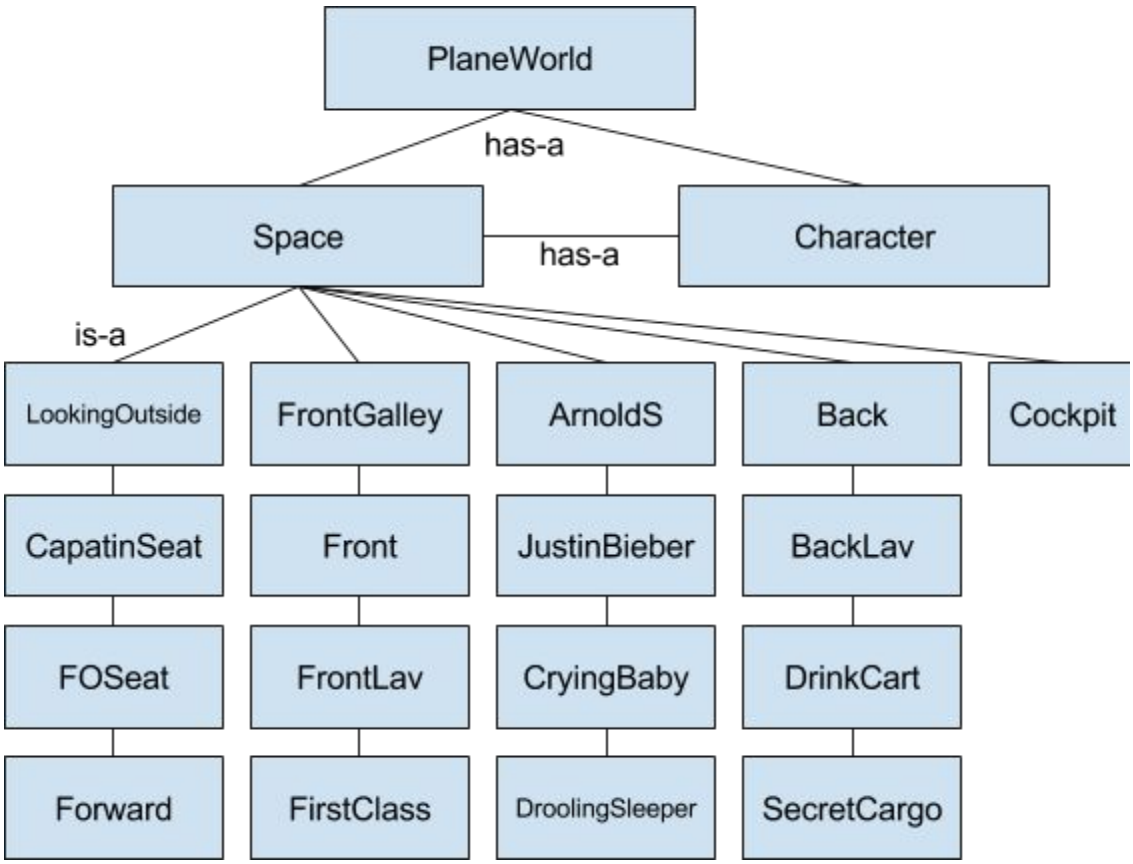
Cockpit - Core Space with forward, back, left and right pointers. Locked until character gains access

Captain Seat - Space pointed to by Cockpit where interaction occurs

First Officer Seat - Space pointed to by Cockpit where interaction occurs

Looking Outside - Space pointed to by Cockpit where interaction occurs

Outline



Game Play

The following steps explain the solution to winning the game. *talk()* and *lookAround()* enhance game play for the user by providing clues, but *manipulate()* is the only function necessary to move through the game

1. Character starts in Back Lav. Secret cargo area and Cockpit are locked.
2. Move two spaces right to the Drink Cart and get the energy drink from the flight attendant to wake the drooling sleeper.
3. Move left, then one space forward to Economy
4. Move right to the Drooling Sleeper and give him the energy drink. Gain candy from the Drooling Sleeper to stop Crying Baby
5. Move two spaces left to Crying Baby and use the candy. Gain Selena Gomez's phone number.
6. Move one space right and one space forward to First Class.
7. Move one space right to Justin Bieber. Give him Selena Gomez's phone number. Gain forward flight attendant's name
8. Move two spaces left to Arnold Schwarzenegger. Give him forward flight attendant's name and gain an autograph.
9. Move one space left and one space forward to the Forward area.
10. Move one space right to the Front Galley. Give flight attendant the autograph to gain a key for the Front Lav secret hatch
11. Move two spaces left to Front Lav. Use key to gain secret knock for access to the cockpit.
12. Move one space right and one space forward to the cockpit.
13. Move one space right to the First Officer. Gain coffee for the Captain
14. Move two spaces left to the Captain. Give him coffee and gain key for the Secret Cargo area
15. Move one space right and all the way to the back to the secret cargo area.
16. Talk with Keeper in Secret Cargo area. Answer to riddle is "Nothing"
17. Game ends.

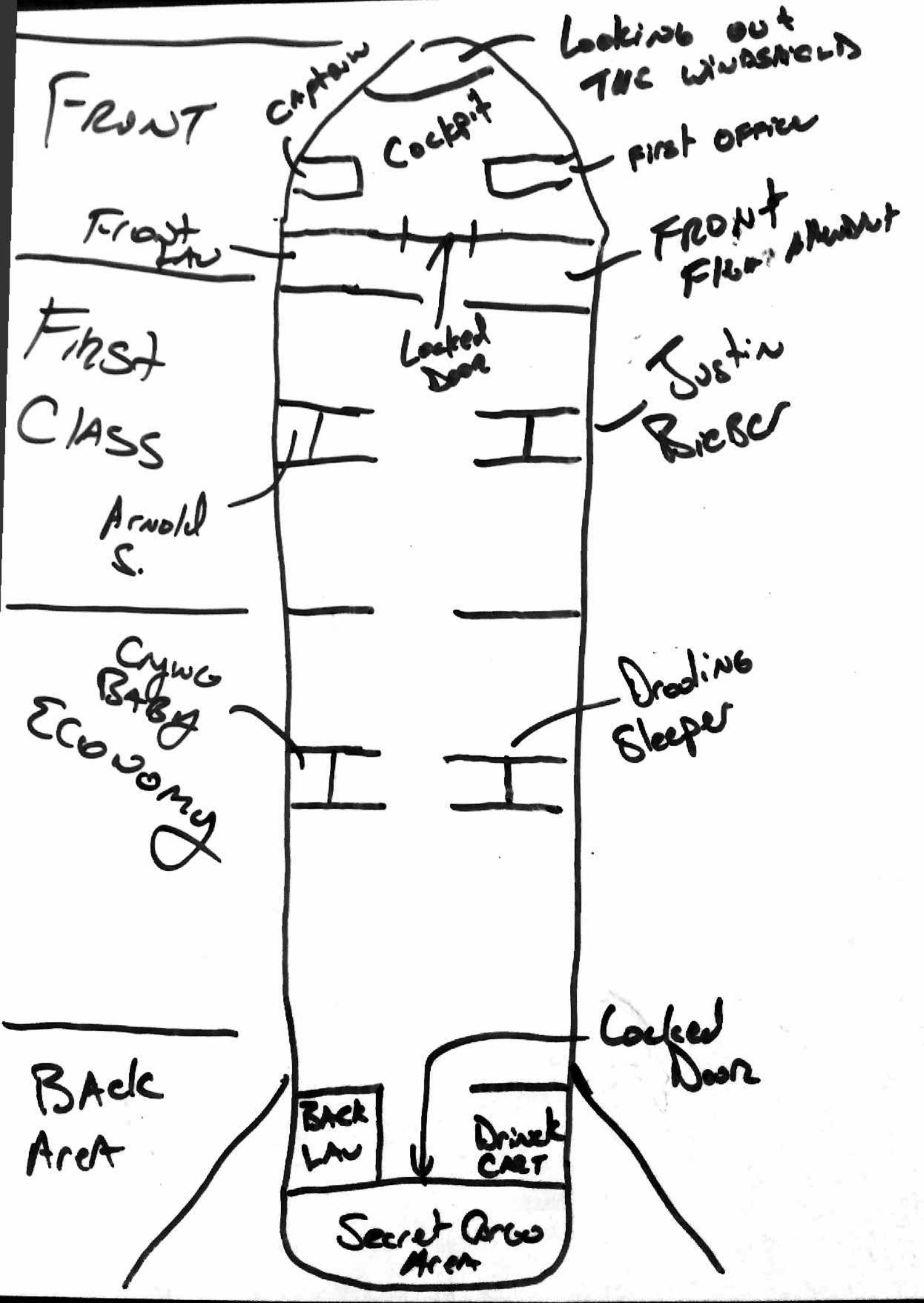
Reflection

Creating this game was a great learning experience. The most difficult problem I ran into was figuring out how to properly move the character between spaces. My original plan was to have a *charPresent* boolean value that would be set to *true* as the character moved to each Space,

however, I found that it was much easier to have every space point to the same Character object, and change the *currentSpace* value in the Character object accordingly.

Space contains a Character object member variable, and Character a Space member variable. This wouldn't compile at first. I learned that the header files had to be imported into each class's .cpp files with a forward declaration in the .hpp files to achieve this.

Booleans were added to the Character object to represent the keys for locked doors, and logic added to *moveFwd()* and *moveBack()* to prevent movement into these areas without *hasKey* and *hasCKey* being set to true. Originally, I had added these booleans were part of the Spaces for which access was prevented.


FRONT
Captain
Looking out THE WINDSHIELD
Cockpit
First Officer
Front Left
FRONT Flight Attendant
First Class
Locked Door
Justin Bieber
Arnold S.
Crying Baby
Economy
Drooling Sleeper
Locked Door
Back Area
Back Lav
Drink Cart
Secret Cargo Area