Ryan Adams
12/10/2019
CS 162 – Final Project

main() Pseudocode

```
Create start menu that asks player if they want to play or quit
Loop that will continue to play until the player wants to quit
    Create game map
    Display scenario and directions
    Pause then clear screen
    Print beginning map and patience
    Loop that continues until patience < 1 or player is sitting
        Ask player what action they want to take
        Switch based on action choice
            If player wants to move, ask direction
                Move in specified direction
                Clear screen
                Re-display updated map and patience
            If player wants to check items
                Display items
                Pause and clear screen
                Re-display updated map and patience
            If player wants to interact
                Interact based on space
                Pause and clear screen
                Re-display updated map and patience
    If player is sitting
        Display winning message
    Else display losing message
    Evaluate if user wants to play another game and break loop if they want to quit
Say goodbye
```

Ryan Adams
12/10/2019
CS 162 – Final Project

Reflection

Polymorphism

The polymorphism was very similar to projects 3 and 4. I used an abstract Space class and created 3 derived classes with different characteristics. Notably, the interaction was different for each space class. I did have some trouble finding a way to have one derived class interact with a different kind of derived class. I figured out a way by having a common function within the Space parent class.

The Game

I decided to make a funny, "boring" simulator as my game. I thought it would be fun to be the guy in the plane that is always disrupting other passengers on a quest to get comfortable. To do this, I set the goal as something mundane. In this case, the player just needs to put their luggage away and find a seat. The obstacles come in the form of needing to find an empty overhead bin and then finding an available seat. The "enemies" in this case could be the passengers, who will be annoyed if you interact with them poorly and the stewardess, who is constantly watching you and just waiting for you to mess up. The "health" in the case is the stewardess's patience with you. If she becomes inpatient, she has the authority to kick you off the flight.

Items

I decided to make a pretty simple Item bag class that just stores up to 5 items in an array. This is because the items do not necessarily need their own functions. The items act almost like keys. You start with a "key" in the form of luggage, that needs to be put in a "lock" in the form of an empty overhead bin. Then you find other "keys" along the way in the form of other airplane items, which again need to be put in a "lock", which in this case is the correct person. Only then will the goal, which is an empty seat, be available to the player.

Troubles

I did have some trouble with an error that caused some crashes. It seemed to be due to making too fast an input after the program pauses. I had put in system calls to both pause the program and wait for an input and to clear the screen. This was in an effort to make the user interface more approachable. It would pause on important information, like the instructions or the item list display. Once you press a button, it would then clear the screen and move on. The issue only seems to happen when you press too many keys at once during a pause. I was unable to solve this, but I did determine that it had something to do with my askForInt function. I believe it was reading the extra inputs in the buffer from typing too fast and then causing an error.

Ryan Adams
11/10/2019
CS 162 - Project #3

Testing

main()

Test Case	Input Value	Expected Outcomes	Observed Outcomes
Action: Move to Aisle	Move to Aisle	Display aisle message re-display updated map	Display aisle message re-display updated map
Action: Move to Seat	Move to Seat	Display seat message re-display updated map	Display seat message re-display updated map
Action: Move to Person	Move to Person	Display person message with random problem re-display updated map subtract patience	Display person message with random problem re-display updated map subtract patience
Action: Interact with Aisle	Interact with Aisle	Check random status of overhead if empty and have luggage, put in luggage if it has item, take item	Check random status of overhead if empty and have luggage, put in luggage if it has item, take item
Action: Interact with Seat	Interact with Seat	Attempt to sit in seat if still have luggage, display negative message if luggage stored but seat unavailable, display negative message if luggage stored and seat available, display positive message	Attempt to sit in seat if still have luggage, display negative message if luggage stored but seat unavailable, display negative message if luggage stored and seat available, display positive message
Action: Interact with Person	Interact with Person	Ask item to give If no items available, display empty item message if person does not like item, display negative message if person does like item, display positive message and make adjacent seat available	Ask item to give If no items available, display empty item message if person does not like item, display negative message if person does like item, display positive message and make adjacent seat available
Action: Check items	Check items	Displays Items	Displays Items
Action: Check items w/ no items in bag	Check items w/no items in bag	Displays empty message	Displays empty message
End turn sitting	player sitting	End loop and display winning message	End loop and display winning message
End turn with patience < 1	patience < 1	End loop and display losing message	End loop and display losing message