Adam Wright

CS-162
Final Project: Design and Reflection

Project Plan:

For the final project my plan is to create a rat in a maze game. There will be six rooms that inherit from the base space class called space. I want a map to be displayed alone with some old school ascii art that will make each room feel unique. The base class space will have 4 pointers as the assignment requires and there will be a setter method in space that will allow me to figure out the order of the map by entering in another derived room or null. This should make it easy to set up the map once the classes are set up. I want each turn to print the room's art and the map and also a meter that shows how much life the player's rat has left. The player will be able to move in the four cardinal directions for each turn and if they don't read the map and select a wall then NULL will be found and they won't move. Each turn the player's health bar will go down by one. They will win the game by finding a key and then reaching the final door with the key.

Pseudocode:

do while loop wrapping the game to allow restarting and exit

cout rat ascii art

entering 1 continues to the starting maze room

entering 2 will exit the application

choose 1-4 to move in a cardinal direction

room's action runs

print room image

print inventory

print cheese o meter

print map

choose 1-4 to move in a cardinal direction

find key and navigate to finish on map

use key to exit

print win or death

while loop returns to beginning play or end menu

Testing Table:

Testing Input	Expected Output	Actual Output
run program main	do while prints rat and menu	do while prints rat and menu
User enters 1 to enter game User enters 2 to exit program	prints directions exit and free memory	prints directions exit and free memory
move into wall each step health meter 0 3 items already each step	warning and no new item cheese o meter loses 1 each step grim reaper and game ends told that items are full map renders characters position	warning and no new item cheese o meter loses 1 each step grim reaper and game ends told that items are full map renders character's position
room 1 named start press enter in start room room 1 room 1 hit wall	stick item prompt print image/map/inventory: stick/menu able to move up or right move and return get new stick no new stick	stick item prompt print image/map/inventory: stick/menu able to move up or right move and return get new stick no new stick
empty room	no item	no item
dinosaur room has stick no stick greater than 3 health no stick less than 3 health hit wall cheese room hit wall	oh no prints don't loose 3 health loose 3 health death message game ends no new attack recieve a piece of cheese no new cheese	on no prints don't loose 3 health loose 3 health death message game ends no new attack receive a piece of cheese no new cheese
key room hit wall	receive a key no new key	receive a key no new key
door room door room	no key then can't move into finished told that you can finish	no key then can't move into finished told that you can finish
finished room 14 press enter in finished room	game win start game menu	game win start game menu
start game menu	play again or quit	play again or quit
User enters 1 User enters 2 to exit program	start a new game exit and free memory	start a new game exit and free memory

Reflection:

This project was a lot of fun in the end. I enjoyed looking for quirky ascii art that would bring some life to the game and make it more interesting to look at. One of the problems that I ran into was trying to print the map with a for loop. I rendered the Cheese o meter health bar with a for loop, but then I sat trying to think of how I could make the map work in the same way and then I realized that I should just move forward and set a map for each room and then just use a virtual print function to render the map as a string. I also ran into some trouble in figuring out why my game loop would only allow be to exit on a win and not on a death. I found that I needed both the outer and inner game while loops to check for death to make sure that I actually could die in the game. I ended up adding a number of other conditionals as I went such as one to exit the game loop early if the dinosaur killed the character so that the game wouldn't run another useless round before the death was shown. I wasn't looking forward to this program when I first looked at the requirements, but in the end I have come to like it the most of all and I have to say that it is a good way to end the term.