Final Project: Design, Test Plan and Reflections

Tucker Shannon CS162 Final Project

Project Description:

Make a Text-Based Game that uses good object oriented program style, including inheritance, polymorphism, and pointers. You will create a series of rooms for the player to move through. The rooms must use four pointer variables for movement. You must have at least 6 spaces of at least 3 different types. You must have an abstract space class. The player must have some way to keep track of items such as a backpack or inventory for things such as a key. You should be able to interact with structures.There should be something like a time limit implemented.

Design Plan:

This term I've had a more reasonable final schedule and was able put more time towards this project than usual. As a result I want to make this game as interesting and challenging (to myself) as possible to see what I can come up with. First, I'd like to do away with my method of input that I've used in previous labs and projects and use one that takes keyboard arrow keys instead. I'd also like to make a series of games in each room to create a fun, fast paced game. I will try to implement a series of arcade style games for each room.

Type of game room ideas:

- Space Invaders style monster shooting
- Falling ceiling dodge game
- Flappy Bird/Helicopter style Game
- Dance Dance Revolution Game
- Maybe blow up the Death Star

In each room I will have a different task or game

Room class:

This class will act as my abstract space class. All my rooms will inherit from this class and will have pointers to other rooms within this class. The game will run from inside this class

NPC class:

This class will have my characters and enemies functions. It will be an abstract class that I will use polymorphism on to create enemies and players. Example: icicles falling from the ceiling will be a form of "npc"

Player Class: Inherits basic function from NPC class but also has an inventory to keep track of coins left and keys for the next game and so on.

Menu Function:

This function will display controls to the user and call games for them to play.

Game Progression:

1. Get bag of quarters to play the games
2. Shoot the aliens descending from space
3. Dodge the falling icicles from the ceiling!!
4. Dodge the cave walls as you fly through the narrowing cave
5. Use the keyboard to dance your way to victory with dance dance revolution
6. Final challenge: Blow up the death star!!!

Test Cases:

Test	**Result**	**Expected Result/Adjustment**
Start the game but press random keyboard input (garbage input)	First:skipped steps After: no errors	Made it so only the enter key will progress instructions
Use arrow keys to go to undesired places	Nothing, because I've limited the use of arrow keys to fit the application	Nothing
Character attempts to leave room walls	Nothing, motion is limited to within the wall boundaries to prevent cheating	Nothing
Run out of game coins (quarters)	Can no longer continue playing the game. Game over	GAME OVER
ALIEN GAME: Don't shoot any aliens.	Aliens conquer your base. Must retry, with a game room restart. -1 QUARTER	-1 QUARTER
ICE GAME: Get hit by ice	Insufficient points to	-1 QUARTER

before getting at least 15 points.	advance. Must get at least 15 points -1 QUARTER. Game room reset	
CAVE GAME: Hit the wall before getting at least 6 points	Insufficient points to advance. Must get at least 6 points. -1 QUARTER. Game room reset	-1 QUARTER
DANCE DANCE REVOLUTION GAME: Miss too many notes before 50 points	Insufficient points to advance. Must get at least 50 points. -1 QUARTER. Game restart	-1 QUARTER
Move past a boundary	Not possible	Nothing

Reflections:

Problem 1: I tried to get keyboard arrow key inputs with iostream with no success so I switched to c++ curses for input instead. I don't know why we don't use ncurses in place of iostream to start. It seems much more capable, easy to use, and much much more powerful. I was very happy to find this!

Problem 2: The screen would only update when I pressed a key. I didn't like this because I would like to set a game "pace" so things much without having to have input. I find out ncurses has a "timeout" option on input that will skip keyboard input and continue the loop if nothing is pressed in time.

Problem 3: The movement of the cave walls were pretty difficult to do. I had the walls sort of randomly spawning without linking nicely. I had to be very careful with coding to get the walls to follow each other through their movements.

Problem 4: I didn't take care of memory leaks until I had pretty much finished everything. Definitely should have taken care of leaks as I wrote the code so I didn't have to backtrack so much.

Problem 5: I wanted the final room to have some cool animation. I wasn't able to print text blocks to the screen so I had to write a simple function to print .txt files to the screen. This took more effort than I liked but it looks pretty cool when the death star explodes. PS. The animation for the explosion was NOT done by me. I included a link to the original creator in the finalRoom.cpp comments.

Final Thoughts: I really liked that they allowed a more open-ended problem. I realise this is much more difficult to grade but I can’ tell you how much I appreciated being able to think of things to create. This was a very fun project! If I were to do it again I would probably spawn more than 1 alien in the the first challenge, add gravity to the cave challenge, add some animation to dance dance revolution (maybe so MIDI music too lol). Also I would have like to clean my code up a lot. It could definitely be written much cleaner but I think I’ve pushed my other class off too much already..