

Andre Ortega

CS 162

Final Project Plan, Test, Reflection

PLAN

Prompt the user for entry on the main screen

Display all instructions

Create an array of pointers pointing to array of space pointers

Allocate all pointers to necessary Space Types

- Set position of space w/ golden idol
- Place a big cavernous hole in the way
- Randomly place rocks, traps, hole, and blank spaces throughout
- Disguise trap spaces as blank squares

Run through entire 2D array and set all pointers to necessary variables

Define the Indiana Jones class

- HP w/ set/getHP
- Current x/y coordinates
- Direction to check if able to move

Place Indy on the board

Begin simulation

- Indy moves with W-A-S-D + [ENTER]
- Board cycles properly
- Hidden traps trigger HP loss from random roll
- Indy's direction will check which way user is trying to move
 - Ensure Indy cannot walk into rocks
- Move Indy to blank space if he walks into a hole, updating HP with fall damage
- Ensure Indy can only cross cavernous hole using the whip, rewriting Indy XY coordinates
- Same upon return
- Indy is able to use Items in his Linked List
 - Water heals once

- Bag of sand is used to retrieve idol
- Whip is used to cross cavernous pit
- Other items print messages and have uses in certain situations if time allows
- Once Indy is at the IDOL space, Bag of Sand can be used to retrieve IDOL
- IDOL is added to the inventory
- If Indy can return to the entrance with the IDOL, the player wins.

TEST CASE	INPUT	EXPECTED OUTPUT	OUTPUT	REMARKS
Start game	'1'	[game begins]	[game begins from menu]	As expected
Generate/print board	[start game]	[board is printed w/o leaks or invalid referencing]	[board prints]	Expected output
Print instructions	[start game]	Instructions are printed	Instructions are printed	Expected Output
Access items	[i]+[ENTER]	List of items printed	List of items printed	Expected Output
Fall in hole	Navigate to cavernous hole	Indy takes damage, and is moved to start	Damage is taken and Indy is moved properly	Expected Output
Find a trap	Walk until a trap is found	Indy takes random damage in proper range	Indy takes proper damage	Expected Output
Use Water	Access water item	Indy s healed by 10 pts	Indy is healed properly	Expected Output
Use Whip	Use whip to cross hole	Indy is transported	Message displays/Indy transports	Expected Output
Use hat/sidearm	Navigate to both items	Cheeky output	Cheeky output is displayed	Expected Output
Kill Indy	Just kill him	Game Ends, player Loses	Game ends	Expected Output
Test ALL borders	Run into everything	Nothing catastrophic occurs	Game doesn't quit, proper error messages are printed	Expected Output
Grab the idol!	Go get it!	Game behaves itself	Idol is added to inventory in place of sandbag	Expected Output
Win the game	Grab idol and run	I win.	I ran into a hidden trap and died	Game suks

Try winning again	Grab n' go	I win.	I win this time	Expected
VALGRIND	Play the game, quit	No leaks (crosses fingers)	Oh look a leak	Search and destroy the leak
VAGRIND	Play again, quit normally	Noleaksnoleaksnoleaks	No leaks!	Leak occurred at updating linked list w/ idol, bag o' sand was not deallocated upon switch
Play more with valgrind	Do it all	Game acts appropriately	Everything works	Expected Output

Reflection

What a fun doozy. I decided to make all items and space types generated by dynamically allocated pointers, feeling like I finally have a decent handle on their use. Updating the linked list to use pointers to items wasn't too difficult, however, remembering where they needed accessing and deletion was important to remember as it took me a moment to find that memory leak that gets left behind by the Bag of Sand.

I estimate about almost half of this assignment was borrowed directly from previous assignments, the rest was pretty fun to implement now that I have a good understanding of how most errors are made, and how memory is allocated. Overall, I was able to implement the program exactly as the plan specified, but if I had more time I would make better use of the items, and cause the cave to begin collapsing after the idol was retrieved by rolling for every blank space to become a rock space. I would have also found a way to use that trusty sidearm if time allowed.

I learned a lot this quarter, I appreciated your input throughout. Happy Holidays!