Christian Roccanova

CS 162 Final Project

6/6/2017

Final Project Reflections

The goal of this project is to create a game that incorporates various ideas learned throughout the course. The game will be designed as a cave or dungeon represented by a board in which the player will give input to move up, down, left or right. They will have to collect a set of keys to unlock doors leading to a monster which they will need to defeat. In order to defeat the monster, they will also need to collect a magic sword and magic shield. To this end, a Space parent class will be created which will contain virtual functions for each child class to override. The virtual functions will include printBoard() to print the current room, move() to move the player token, isWall() to check if the player is trying to move into a wall, door() to move between rooms and specialSpace() to activate the special spaces within a room such as a key. The child classes will include MainRoom, SwordRoom, ShieldRoom, KeyRoom, BossRoom. The main file will contain the instructions and a menu() function to allow the player to play or quit the game. Selecting play will call the play() function of the Game class which will run the game as a whole including taking inputs and calling functions from the other classes. An inventory class will be added to hold the items in a queue until they need to be used.

Testing

Test	Expected Outcome	Observed Outcome
Enter an option other than 1 or 2 on the main menu	Displays an error message and asks for a new input	Displays an error message and asks for a new input
Enter an option other than w, a, s or d in the play function	Displays an error message and asks for a new input	Displays an error message and asks for a new input
Move into a wall on each type of room	Displays a message that the player bumped into a wall and does not move the player token	Initially displayed the error message, but still moved the player – this was due to a syntax issue in the line testing if the next space was a wall or not
Move into the moat in the main room prior to activating the switch	Displays a message that the player cannot cross and does not move the player token	Displays a message that the player cannot cross and does not move the player token
Move into the space that used to be the moat in the main room after activating the switch	Player should move normally	Moves normally
Moves into the doors from the main room	Moves player to the corresponding room	Initially moved through the door, deleting it, this was due to a similar issue as noted above in the wall test and was promptly fixed.
Moves into the monster door	Displays a message that the door	Displays a message that the door

before hitting the key switch	is locked and does not move the player token	is locked and does not move the player token
Moves into the monster door after hitting the key switch	Moves to Boss room	Moves to Boss room
Move onto sword	Adds sword to inventory	Initially failed to retain inventory outside of class – this was fixed after container was moved to Game class
Move onto shield	Adds shield to inventory	Initially failed to retain inventory outside of class – this was fixed after container was moved to Game class
Move onto Switch room switch	Replaces moat with normal blank spaces	Initially did not replace moat due to being misplaced in the order of function calls, but was fixed after being moved.
Move onto Key room switch	Unlocks door to Boss room	Unlocks door to Boss room
Move into monster with both sword and shield	Game displays victory message and returns to main menu	Game displays victory message and returns to main menu
Move into monster with only either sword or shield	Game displays defeat message and returns to main menu	Game displays defeat message and returns to main menu
Move into monster with neither sword nor shield	Game displays defeat message and returns to main menu	Game displays defeat message and returns to main menu
Move a greater number of moves than the turn "timer"	Game displays defeat message and returns to main menu	Game displays defeat message and returns to main menu

Due to an initial misunderstanding of the requirements of the assignment, the concept of keys was changed to be two separate switches, one to unlock the boss room door, the other to remove a moat blocking that same door. This included adding a SwitchRoom child class to the program. When initially added, the switch was not removing the moat as intended, but this turned out to be because the function call for checking if the switch was activated was being placed in the wrong spot. During initial testing there were issues with the walls and doors where the message that a door or wall was walked into would display, but the player would still move into that space. This turned out to be an issue with the syntax of the test checking whether the space was in fact a wall or door and was quickly corrected. The sword and shield initially had issues with the inventory class as the inventory object was placed within the Space parent class. To fix this, the inventory class as a whole was removed and replaced with a standard template library queue stored in the game class. Finally, getX and getY functions were added to several of the child classes in order to return the x and y coordinates of the player token within the respective rooms.