Shayna Snyder

CS372

25 January 2017

Final Project: Final Deliverable

For my final project, I created a single-player game that merges the board games Clue and Mastermind. This is similar to Clue in the game’s overall objective and the way the board. Like Clue, the objective of the game is to guess the correct suspect, weapon, and room. The game implements aspects of Mastermind in order to make the game single-player. For each guess the user makes, they are informed of how many items are incorrect and correct, but not which items are which. The correct combination of the three objects is randomly generated. Then the user makes guesses, and through a series of functions, the game outputs information about their guesses.

My project specification has changed a lot throughout the process of making this code work. Mostly, the specification has just gotten very bulky with a ton of variables (some of them are likely superfluous.) I made a GameTest class to hold my resources and my JFrame. I wanted to make my GameTest code short and sweet, so I wrote the bulk of my code in my Game class. Then I added a new Game object to my JFrame in my GameTest main function. I created three more classes to hold my suspect, weapon, and room objects.

With so much code, a lot of the challenges I faced probably could have been solved with clearer commenting. But due to the amount of variables and functions I have in my Game class, there were many mix-ups with calling different functions and reassigning variables. Another challenge I faced was with JCheckBoxes. After a lot of frustration with buttons that were not showing up in my JFrame, I learned that JCheckBoxes do not work with a null layout. JCheckBoxes only work in layouts such as GridLayout. I resolved this by adding my boxes to a JPanel with a GridLayout, then adding the JPanel to my JFrame. I also struggled with adding a start menu for my game. I originally tried adding my menu in a separate JFrame, but I could not figure out how to switch JFrames once the start button was clicked. I resolved this by adding a new JPanel specifically for my menu to my JFrame. I made this work by adding my game textAreas and JPanels inside the start button action listener. I also disabled the button and set visible to false for the start button once it was clicked.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Game | | | |  | |
|  | +iBoard : Image  +iSuspects1, iSuspects2, iWeapons, iPawn, iArrows, currentD, title, background : Image  +dice : ArrayList<Image>  +boxes : ArrayList<JCheckBox>  +correctS, correct, correctR, clickX, clickY, mouseX, mouseY : int  -text, text2, notes, categories : JTextArea  +start, rules : JButton  +panelHolder : JPanel[int row][int column]  -type : String  -name : String  -correct : String[ ]  -isGuilty : boolean | | | |  | |
|  | +promptGuess(Room r) : void  +guessSuspect(Suspect s) : void  +guessWeapon(Weapon w) : void  +checkGuess() : void  +paint(Graphics g) : void  +mouseClicked(MouseEvent e) : void  +getType() : String  +getType() : String  +getCorrect() :String[ ] | | | |  | |
|  |  |  |  |  | |
| Suspect |  | Weapon |  | Room | |
| -name : String  -isCorrect: boolean |  | -name : String  -isCorrect : boolean |  | -name : String  -xCor : int  -yCor : int  -width : int  -height : int  -isCorrect : boolean | |
| +Suspect(String n)  +getName() : String  +setCorrect() : void  +getCorrect() : boolean |  | +Weapon(String n)  +getName() : String  +setCorrect() : void  +getCorrect() : boolean |  | +Room(String n, int x, int y, int w, int h)  +getName() : String  +getXCor() : int  +getYCor() : int  +getWidth() : int  +getHeight() : int  +setCorrect() : void  +getCorrect() : boolean | |