Sofia Echavarria

Group Project: Epic Bagel Clicker Game

For the group project, I worked on adding an accessory/prize bank GUI. The user could buy accessories with the bagels they had earned in other aspects of the game. I also helped with some of the formatting of the final combination of panels since I had experience utilizing box layout and border layout. For the accessory bank, I began by creating the outline of the panels, radio buttons, and images. This was difficult because it had a complex visual design, so I learned how to use box layout which was helpful when dividing spaces evenly among different panels. An issue I faced when completing the outline was that the setSize method was being ignored making different parts of the project weirdly sized. When problem solving with Dr. Frewen, we determined that I needed to use the setPreferredSize() method instead of the setSize() method. This difference helped me finish the first part of the bank GUI since the outline was complete.

The next step was adding the images using my two image panel classes. I knew that I needed one image class whose position would not change for my example images and one image class whose position could be moved using the mouse when placing it over the bagel. The first image panel class with the set position was easy since I had already utilized image panels in that way in past projects. But I ran into many issues with my second image panel class. Honestly, I went in circles trying to figure out how to add different images with different positions on the bagel. Then, Dr. Frewen suggested utilizing arraylists with the positions and filenames. So, when a user "bought" a prize, the position and filename would be added to the array list so when it went through the paint component it would go through every image in the array list.

The final step was to make the action listener for the button add the images for the accessory chosen if the user has enough bagels. So, the program checked to see if they had enough bagels to buy the item and then added the item to the arraylists. Also, I learned how to disable a radio button so the user couldn't buy more than one of the same prize. Then, I set the mouse listener to move the image by changing the positions in the arraylist for the last index of the arraylist. I excluded 0 to exclude the initial bagel in the arraylist.

Overall with this project, I learned a higher proficiency in GUIs and working with multiple classes effectively. I learned how to have multiple images in one panel that you can move, and how to use different layout designs to effectively show images.

Individual Project: Design a Planet GUI

For the planet project, I took the old 2.3 Planet homework assignment we did about "getters" and "setters." I took the idea of creating your own planet and made it into a visual GUI. After completing the group project, I felt more confident in my abilities to code GUIs effectively. So, I began with the outline panels splitting the screen in half with box layout so that the top half was the visual GUI and the bottom half were the questions with radio buttons. This went smoothly since I had learned the nuances of formatting like using the preferred size instead of just setting the size.

Next, I went through each action listener and implemented the various parts using the image panel class and the GUI class. First, I created the initial planet setting the color based on the user's selection. Initially, I had set the colors in an array list where there was a Color being added to the array

list each time the button was pressed. This was overcomplicating my program, so I just made it a color variable instead. Then I added rings by adding a transparent picture of rings and sizing it correctly over the image. Next, I gave the program the capability of changing the mass of the planet. I changed this by having a variable for the size, x position and y position for the planet. So, when the mass decreased the size was smaller and the x and y position changed so that the planet was always centered. I had an issue with the rings because I needed the rings to match the size of the planet. So, I created multiple 3 copies of the ring image with different sizes to be used with whatever mass was indicated. Finally, I added moons to the planet by creating a random color method and using a moon method. Then in the paint component method, I used a for loop to repeat painting moons until the amount chosen was reached.

This project had significantly less issues than the group project because of what I learned in the group project about GUIs and formatting.