Animal Sounds Evolution

1. **Describe the state of the project that you encountered:**

- When we first received the game from the previous group, it had a working main menu with a title and buttons for: Play Game, Instructions, Settings, and Quit. The items displayed on the main menu were all bunched together at the top in a line. When a user selected “Play Game,” four animal images were displayed and you could click on each one until they all disappeared. There was no implementation yet that played sounds or allowed a user to return to the Main Menu.

- Main Menu implemented with text and control flow to either
    a. Play the Game
        i. Game Implemented
            1. No Sound as of yet - will add sound in
            2. Right now, the game starts off with a number of images, and when you click on them, they disappear one by one until there are none left.
            3. Will implement logic to progress game based on if the corresponding animal is clicked for the sound played.
    b. Instructions
        i. Not implemented - will create basic instructions for playing the game
    c. Quit
        i. Quit button implemented - exits the applet
    d. Settings
        i. Not implemented - can implement settings that change the volume of different sounds, etc.

2. **In particular, describe any specification, modularization, or implementation flaws:**

- No comments whatsoever.
- The same code was reused each time when an animal image was created. The developers could have created a loop that would loop through an array of possible images, display them, and add functionality to each such as its corresponding sound, logic checks, and behavior (image appearance and disappearance).
- Imbedded 5 different classes into one class file - the developers could have separated these to make the code neater and easier to read.

3. **Fix anything that requires limited amount of work, and keep with the existing setup if changing would be too time-consuming (describe this and justify):**

- Existing setup has the control flow of the game working; when a picture is selected, it is taken out of the lineup. So the basic structure of how the game is supposed to progress towards a goal is already set up. We just need some sort of control structure to decide whether the picture selected is the right one, so that the game has a point, or challenge, to it.
- The way the images were added to the panel and action event checks were made was very inconvenient. If a developer wanted to add a new animal to the game, they would have to copy and paste an entire block of code instead of simply just added the animal to a data structure like an array or map. It would be too time consuming to gain the knowledge to create a loop for generating images, buttons and action event checks, as well as rewrite the entire program in such a fashion. To avoid having too much of the same code, we will only implement 4 animals. Also the way the game was implemented so that you could not start a new game in the same applet session.

Animal Sounds Evolution

4. **Describe what your group is going to do in order to finish the project and produce an acceptable piece of software (the final product):**

- Separate imbedded classes
- Add instructions
- Add sounds that correspond to the correct images, and generate a random sound each time.
- Animal selection: if correct animal is selected, play “correct sound”. If wrong animal is selected, play “buzzer sound”.
- Game logic - whether the correct animals are clicked and selected, so the game has a challenge aspect.
- Exit point in the game - switch control back to the main menu.

For assessment, we submitted our detailed report, and the complete GIT repository with the source code of the final version of the project.