|  |  |  |
| --- | --- | --- |
|  | Looking for | comments |
| Implemented Features | **Number game:**  User can make guesses: Yes  Feedback to user (high/low):  Feedback to user (success)  Used UI elements:  Used a Layout  **Bug World:**  Bugs  Plants  Animated  The bugs have different behaviours  Images instead of circles  Control the animation  Other | User can make integer guesses. The user can make negative guesses, or massive guesses (they just get to be wrong).  Tells the user to guess higher or lower. This means that they should guess higher or lower, not that their guess was lower/higher than the answer.  There is a victory screen and a loss screen. Both include images on a Chartreusse background.  Used a Vbox, buttons, and a background image.  I don’t know if what I did counts as a layout.  Four types of bugs: Wasps, Flies, bees, and dragonflies (CarnivoreBugs)  Only one type of plant, but there are 3 different types of images that represent them (depends on the size of the plant)  They do move around, but it is very choppy (i.e there are no intermediate frames when they move).  Yes. Bees only eat plants. Flies also only eat plants, but they can move 2 spaces in an update (only 1 if their energy is low). Wasps can eat bees and flies as well as plants. Dragonflies only eat wasps, bees, and flies.  Yes, but this can be disabled by changing the name of the image folder or just the name of the image you wish to disable. The code is laggy when using images, but not when using circles.  The animation can be pause/resumed from the pause/play button. There is a slider that controls the animation speed. There is also a quit button that returns the user to the home screen (ending the animation).  Bug world and the number game are in the same program. This program has a home screen that allows the user to choose which game to go to. Each of these games has a quit button that returns the user to the home screen. |
| Quality of code | **OOP:**  General notes:  Coding plan:  What is the abstraction of your code  □ How have you used inheritance and/or composition  □ Encapsulation of the code  **Testing:**  **When marking we will consider these quality aspects of the code. You do not need to explain this but be aware of it.**  - Indentation is correct  - variables have understandable names  - variable are named consistently  - Avoiding repetition of code  - Used appropriate comments | The bug world section of the code is an adaptation of code written earlier in the course. The background behaviour of the bugs, plants, and world are all based on that. The animation behaviour was added after all of that had already been programmed.  The home screen of the program had its own class: main. The number game and bug world each had their own package with a class that dealt with the front end visuals.  Number game had two other classes, which were both event handlers for the buttons (one for the “enter guess” button, one for the “play again” button). (The quit button has its event handler dealt with in the number game class.  Bug world had many classes. It had 2 event handler classes (button event handlers were dealth with in the main body of code) : one for an individual animated entity and one for all of the animated entities.  It also had two different classes for animated entities (one extends ImageView, the other one extends Circle) and an interface to link them together.  In addition to this bug world also had many classes that represent the different types of bugs, represent plants, directions, obstacles, the world, etc.  Each type of bug is its own class. There are two types of animated entities (these are the visuals that represent bus, plant, and obstacles), either they are an animated circle or animated image.  How did you plan and design the programs?  Describe how  What were your considerations about encapsulation?  Write a short comment about how you tested the code. (test method, test cases, boundary cases etc.) |
| Understanding of issues and challenges | We are not expecting comments for all of these questions. Use approximately 300 words and focus on where you have the most important things to comment on.   * How did you overcome issues? * Are there still known issues? Describe them and explain how they can be overcome if you had more time. * Are there aspects what you have done that you are particular proud of? * Are there design aspects you would like to change now that you have implemented it? |  |