How to Use:
To run the program, run the main method of Ninja.java with no parameters. Press the start button to begin the game. Click and drag the mouse to create the slicer to earn points by slicing fruit. Click the button at the top to restart.

Meeting requirements:

Meaningful Object Oriented Design: each type of "fruit" extends an abstract class. They all have certain shared concrete methods such as a run method and a done method. Along with the methods concretely defined in the abstract class they each have their own paint method that they must override. They also have some shared instance variables such as int size, int DELAY_TIME, double xSpeed, double ySpeed, double upperLeftX, double upperLeftY, boolean done, JComponent container.

Event Driven: Using ActionListeners and MouseListeners we can have the user start, reset, and slice the fruit.

Threads: each fruit runs on its own thread.

Appropriate data structures: All fruit objects in existence are contained in an ArrayList so that they may be easily redrawn and kept track of. The modification of this list is synchronized with a lock.

At least one new feature: The fruit is tossed from the bottom of the screen, triggered by an ActionEvent from a Java Swing Timer which we have not studied previously. We also used the mouseCollide method from the MouseAdapter class which we had not previously discussed.

We made sure to maintain good documentation throughout our classes, especially with JavaDoc comments at the beginning of classes and methods. The program did not change much compared to the original proposal, however we did end up using only one panel rather than two because we thought that having a smooth look from the play area to the score and start button was a cleaner final product. We also used the mouseCollide method for collision detection rather than a series of mouse dragged and released checks. The group used github very carefully to collaborate. We were all very cautious to pull all changes to the repository and let others know what class(es) that will be changed before editing anything.