Zain Modi zam374

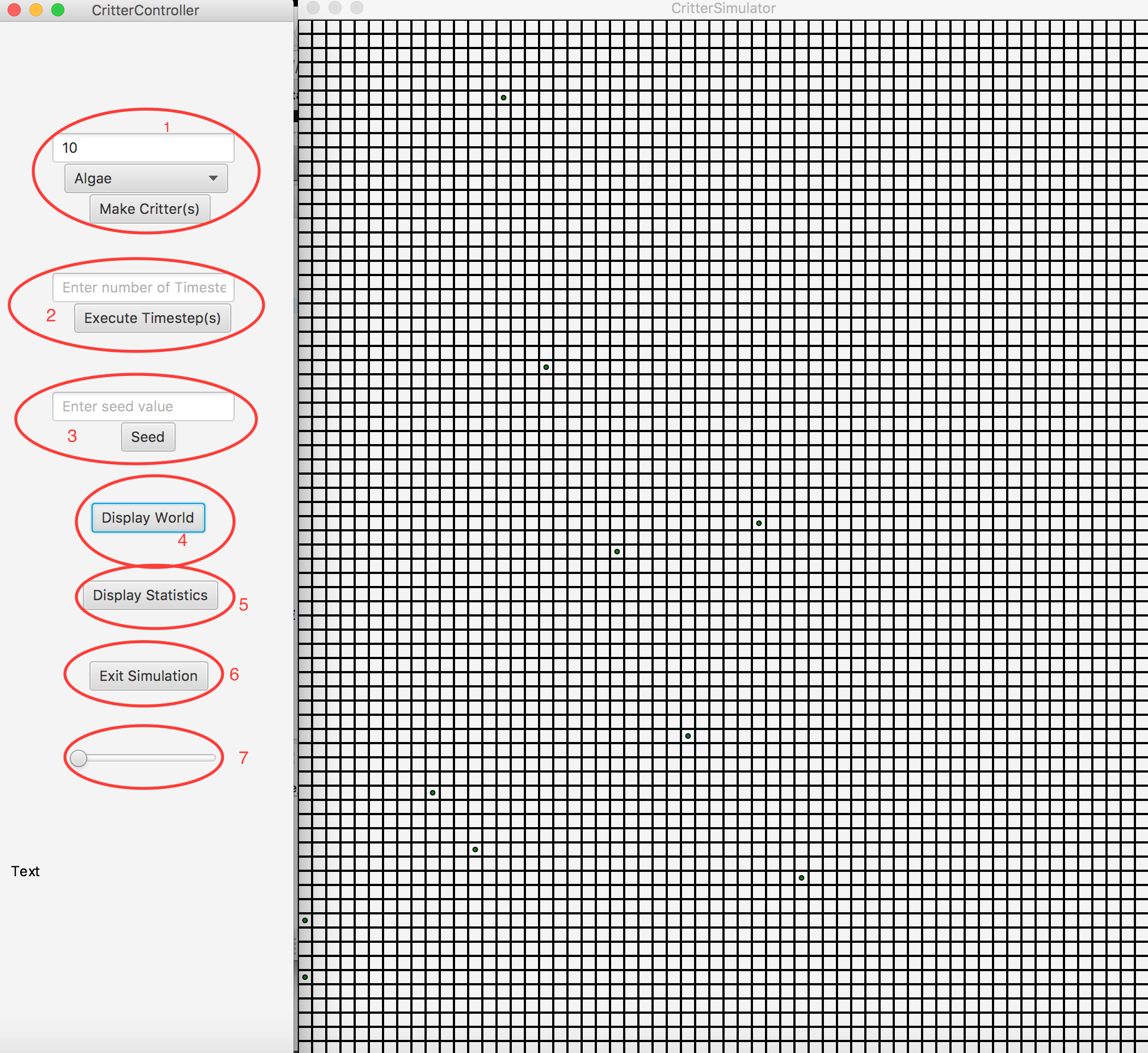
Danny Vo dpv292

**README**

Graphics and Code Description

As far as the code goes in this project, we are mainly reusing the previous game engine that we built, but modified the way it is executed. Instead of it being executed via command line, we

made it execute via GUI. The following is a description of our GUI:



1. These buttons invoke the method

**public** **void** makeCritter(ActionEvent event) **throws** InvalidCritterException

This method takes the event from the button, if an invalid critter was passed(aka the user didn’t choose a critter) an exception will be thrown.

1. This button invokes the method

**public** **void** executeTimeSteps(ActionEvent event)

This method takes the event from the button and executes the amount of timesteps specified.

1. This button invokes the method

**public** **void** setSeed(ActionEvent event)

This method is called when the event occurs and in turn calls a function which has been provided to us called setSeed.

1. This button invokes the method

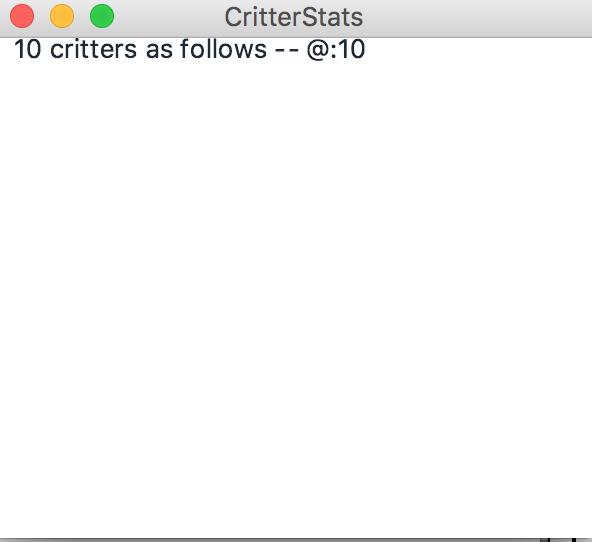
**public** **void** displayWorld(ActionEvent event)

This method is called when the event occurs and calls our show function in Main.java

1. This button invokes the method

**public** **void** displayStats(ActionEvent event)

This method is called when the event occurs and calls showStats in our Main.java file. The showStats method in turn ends up creating a new scene and adding the stats text to that scene. The new scene is shown below:



1. This button invokes the method

**public** **void** endSimulation(ActionEvent event)

This method is called when the event occurs and calls System.exit(0) to terminate the program