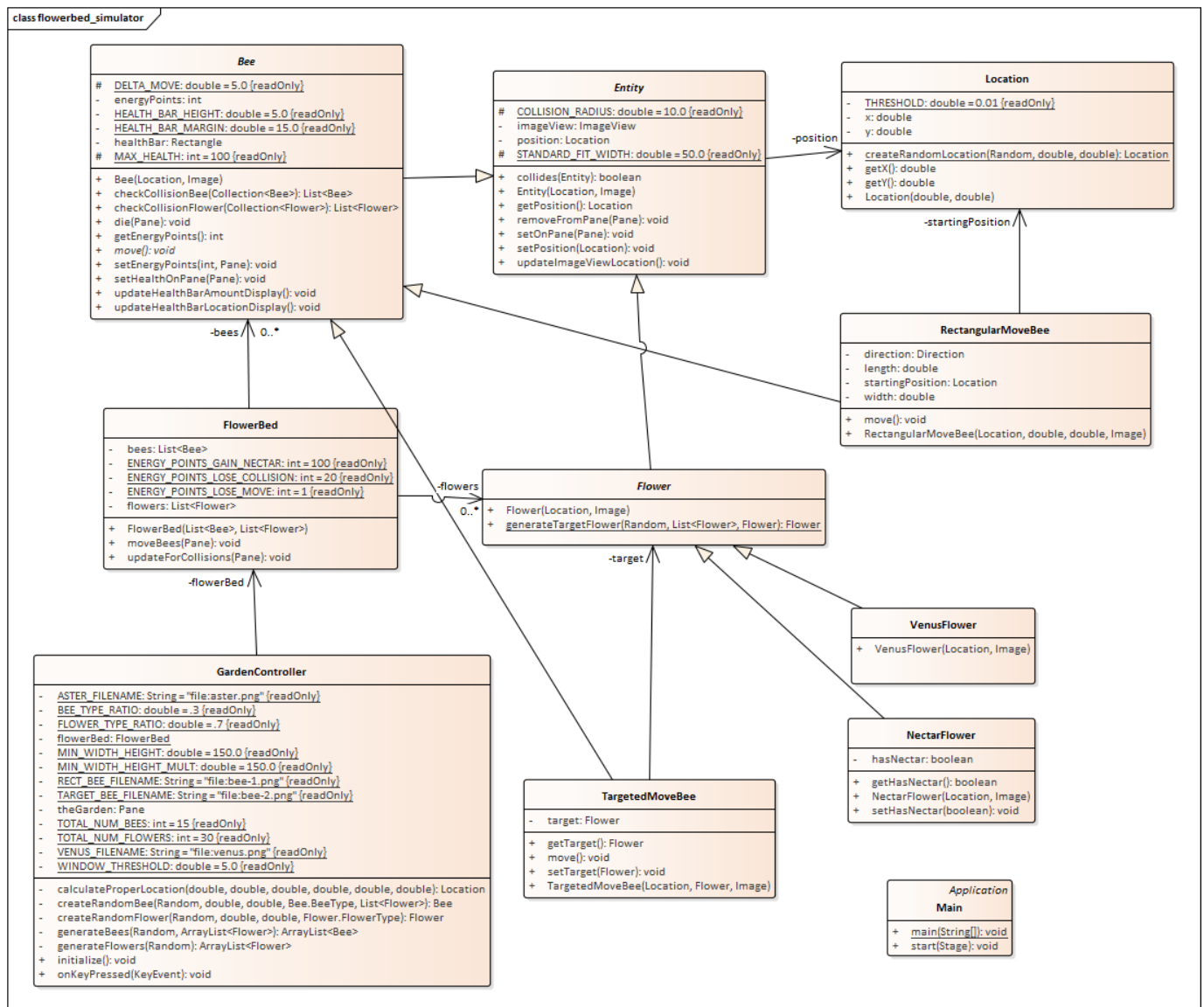


Paul Rinaldi, Stuart Harley
Dr. Hasker
Lab2B Writeup
Submitted: 1/14/2020

Reverse-engineered diagram:

(The .EAP file is also in docs folder in github as well as this PNG)



Significant Changes:

There were several changes from our original design in terms of methods and attributes. Almost all of these had to do with implementing the GUI and correctly displaying the bees. There were no other classes added, besides the FXML, controller, and main classes. We added a

health bar attribute to the bee class, which is simply a rectangle, so there were several new methods dealing with updating the position and size of that.

Who Implemented What:

Stuart Harley:

Base functionality of:

Abstract Entity.java

Abstract Bee.java

NectarFlower.java

FlowerBed.java

Location.java

Full Functionality:

flowerbed_simulator.fxml

Key Press causing movement

Checking Collisions

Generating new target flowers

Paul Rinaldi:

Base functionality of:

RectangularMoveBee.java

TargetedMoveBee.java

Abstract Flower.java

VenusFlower.java

checkCollision

Full Functionality:

Bee Generation

Bee movement

Added/Modified Functionality:

Location.java

Displaying bees/flowers

Collaborated heavily:

Bee.die()

Healthbar

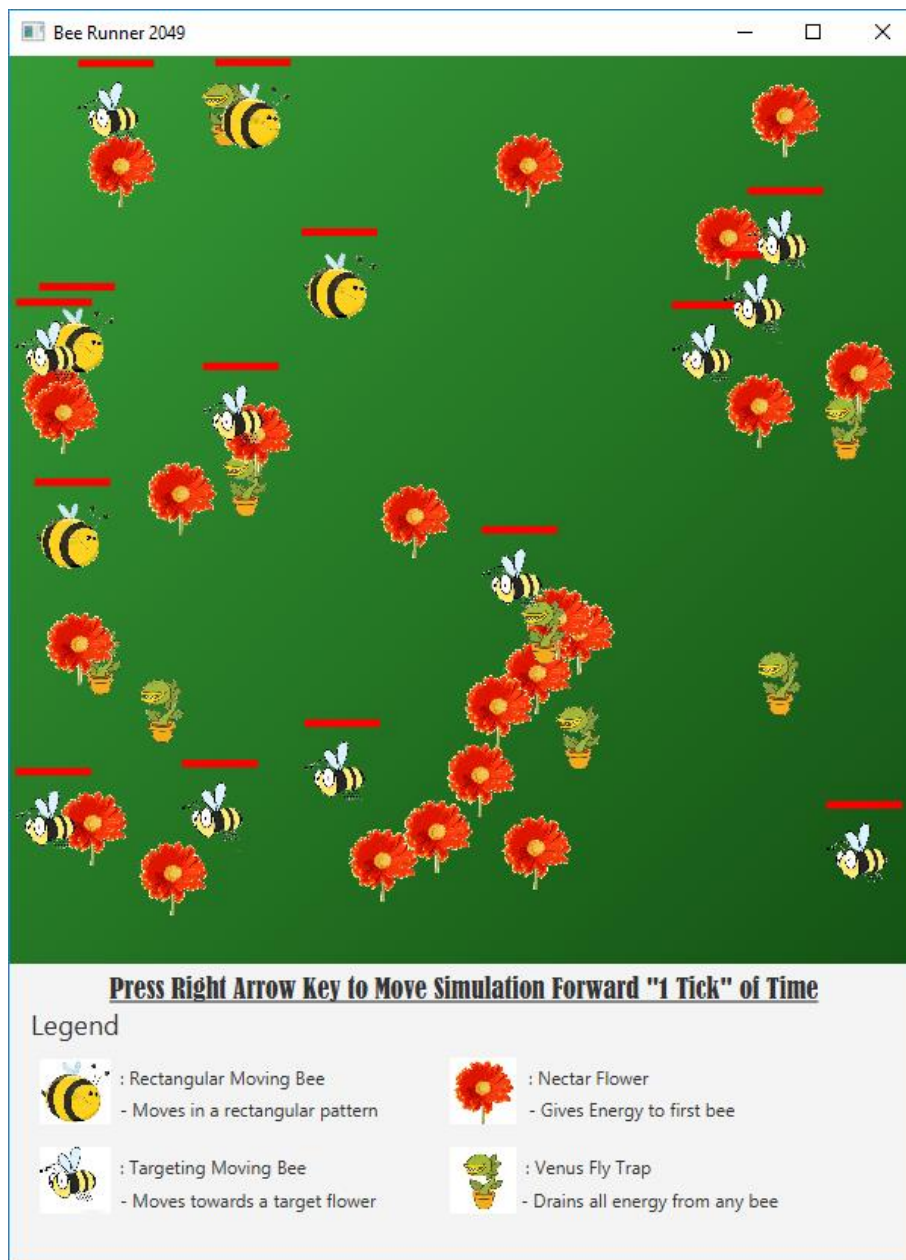
Removing items from panes

Calculating collisions

ImageViews

Displaying bees/flowers

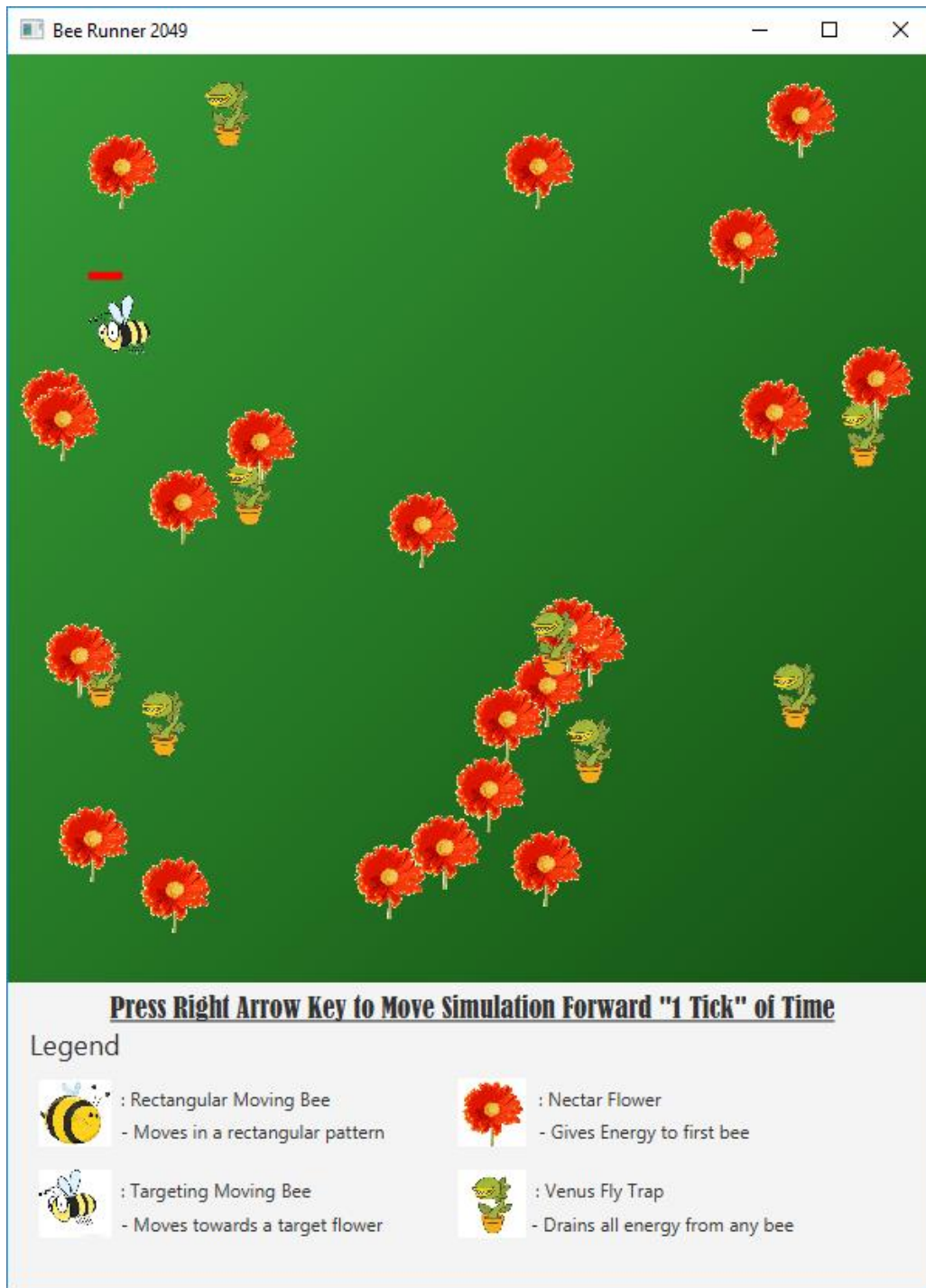
Screen shots of Solution in Action:



Sample Beginning of Program, no bees have moved yet (all have full energy)



Sample midway through program execution, some bees have died out, others are close to dying, and some have gained energy from.



Nearing end of program, one target bee remains.

Known Problems:

There are no known problems in our implementation. There is probably some extra coupling that could have been avoided and there might be less cohesion than there should be with some responsibilities that may not be in the best class.