Andrew Rutherford

CSCI 1300

18 April 2014

1. The purpose of “Universe un = new Universe(zTurtles, z, 600, 600);” is to create a new instance of the Universe class, with a name of “un” and a set of custom defined parameters. These parameters are needed because the constructor in the Universe class requires them. zTurtles is the total amount of turtles (normal turtles + zombie turtles). Z is the number of turtles that start as zombie turtles, when the program is first run. The first 600 represents the width of the canvas in pixels. The second 600 represents the height of the canvas in pixels. This means the canvas has a size of 600x600 pixels.
2. The purpose of the for loop is to set the number of iterations the simulator will run. If you wanted the simulation to run for 1000 iterations, you would change the integer N from 250 to 1000.
3. The Universe constructor takes in four arguments: the number of total turtles, the number of initial zombie turtles, canvas width, and canvas height. The location of each turtle is set by a random variable. “Math.random()\*width” assigns a random value to the x-coordinate of the turtles position, while “Math.random()\*height” assigns a different random value to the y-coordinate of the turtles position. The zombie turtles are created first, followed by the normal turtles. When each turtle is being created, a check is done to see whether the number of zombie turtles created, is equal to the number of zombie turtles that are supposed to be created. If not, that turtle is created as a zombie turtle. The tLocations variable serves as a sort of check to ensure no two turtles move onto the same exact spot on the field.
4. moveZombies determines the direction each zombie turtle will move with each iteration of the simulation. newX and newY are both variables that determine the new x-coordinate and y-coordinate of that turtle. A random value is chosen between the maximum and minimum values of -8 and 8, and that value is added to the existing x and y coordinates of the turtle.
5. The for loop ensures that the number of turtles that move in a given iteration is equal to the total number of turtles on the field. This also ensures that every turtle has moved at least once, before any other turtle moves for a second time.
6. The main for loop in the zombieAttack method is a check to ensure that if any zombie turtle meets the criteria for spreading his disease to another turtle, the disease is in fact spread. The loop runs the same number of times as there are turtles on the board.
7. The radius variable is maximum distance a zombie turtle has to be away from a normal turtle to spread his disease to the normal turtle. The set of logic statements in the zombieAttack method basically find the distance from the top, bottom, left, and right, of each zombie to any normal turtle by taking the difference of the two locations. The x and y coordinate of each turtle are accessed by turtle[x].getXLocation() and turtle[x].getYLocation().

10. The canvas size is set in the constructor method of the Universe class. StdDraw.setCanvasSize(w, h) calls the setCanvasSize method in the StdDraw class, passing in the variables assigned as w and h to be the width and height of the field. The setLocation method in the Turtle class also contains code for the visual display. All of this code calls specific methods in the StdDraw class, depending on what functionality is needed. StdDraw.setPenColor(StdDraw.White) erases the old zombie location. It actually overlays each previous location with a white circle, which when on a white background essentially erases the location. If set to another color, a set of “foot prints” will appear behind every turtle as it moves. Following this is a set of logic statements that assign a color to turtles labeled as zombies, and a different color to normal turtles. StdDraw.setPenColor(StdDraw.RED) sets the color of zombie turtles to red by passing what is inside the parenthesis into setPenColor in the StdDraw class. That argument is StdDraw.RED instead of just “RED” because the RED variable is held as a pre-defined variable within the StdDraw class.

StdDraw.setPenColor(Color.Blue) assigns the blue color to normal turtles. Similar to the zombie turtles, the argument Color.Blue is interpreted by setPenColor in the StdDraw class. Unlike the zombie turtles however, this color is pre-defined in a different class. Color.Blue means the variable is held in the Color class, rather than the StdDraw class. The Color class is one of several of java’s default classes, which must still be called within a class if used. This import exists in the Turtle class at the very top: “import java.awt.Color;”