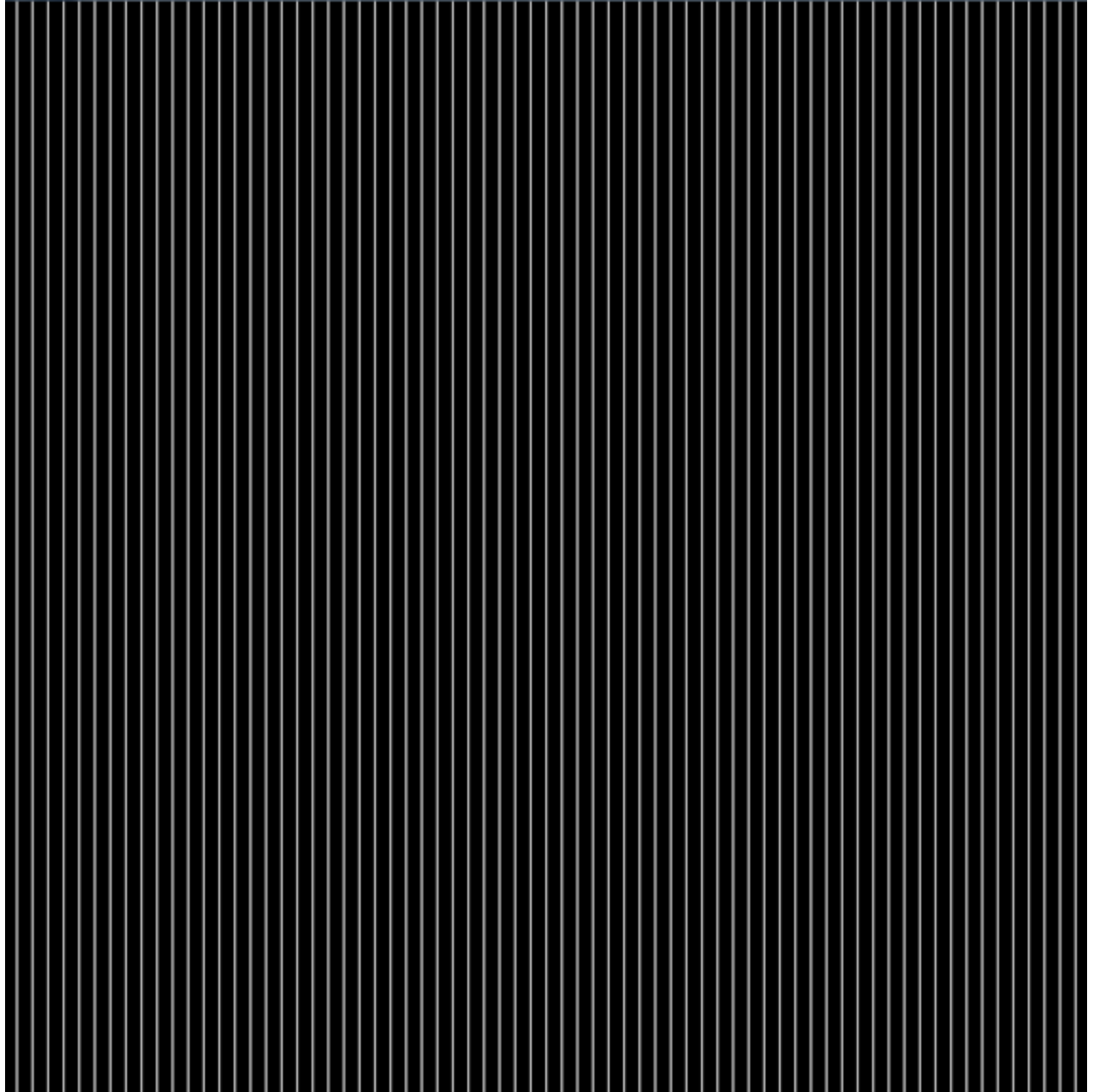# COW - Looping

Sample Problems That We Will Go Over as A Class:
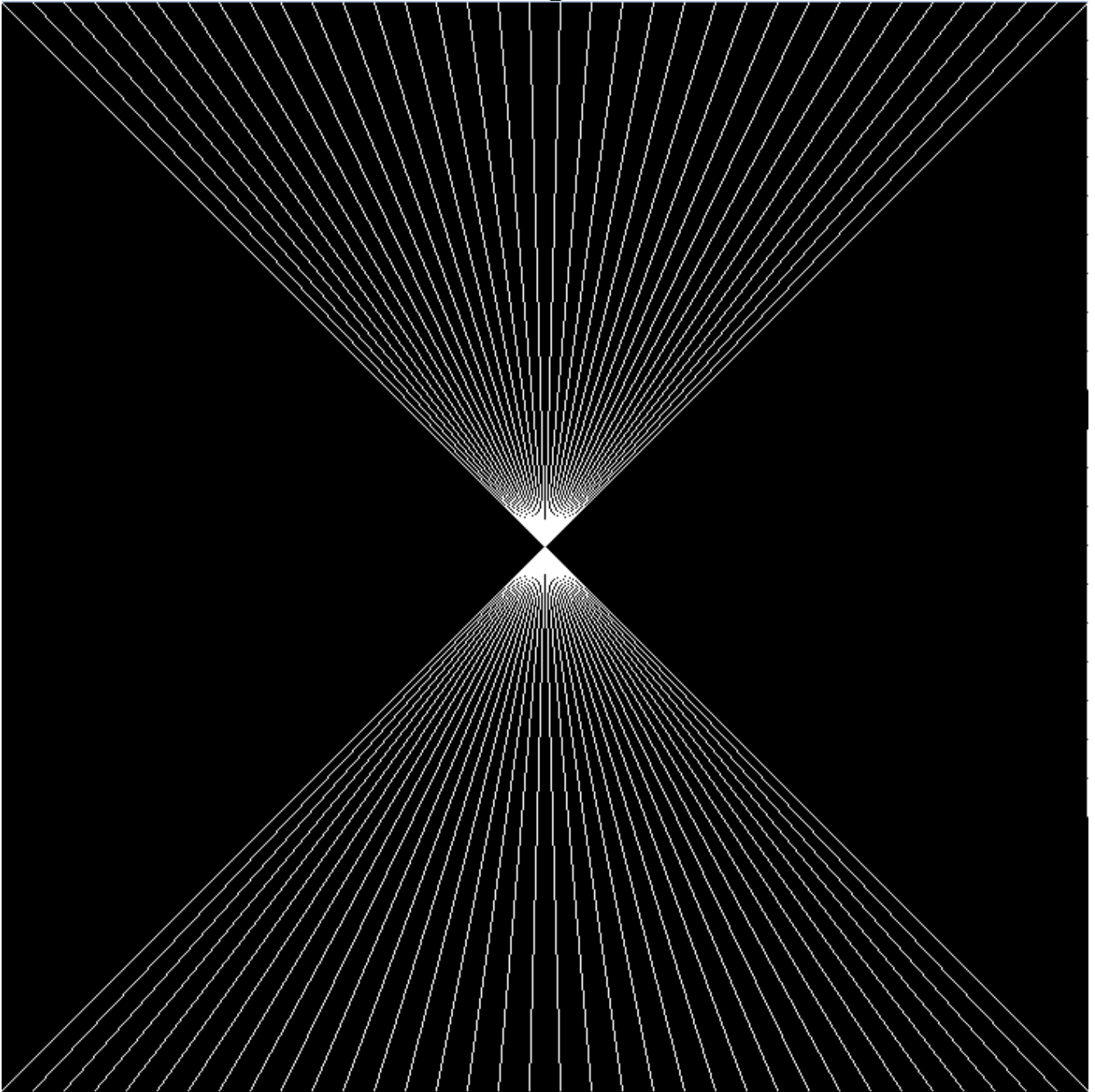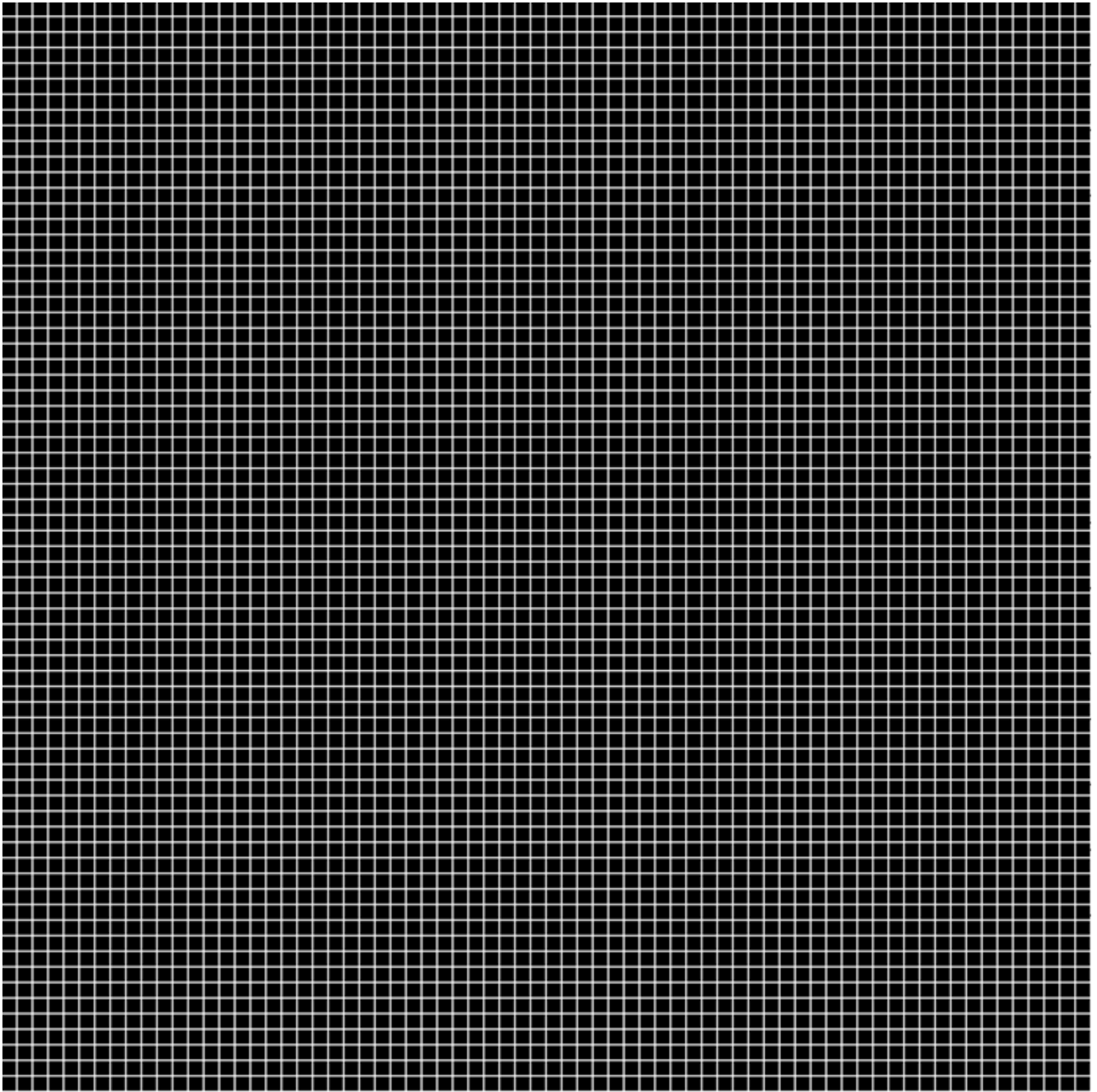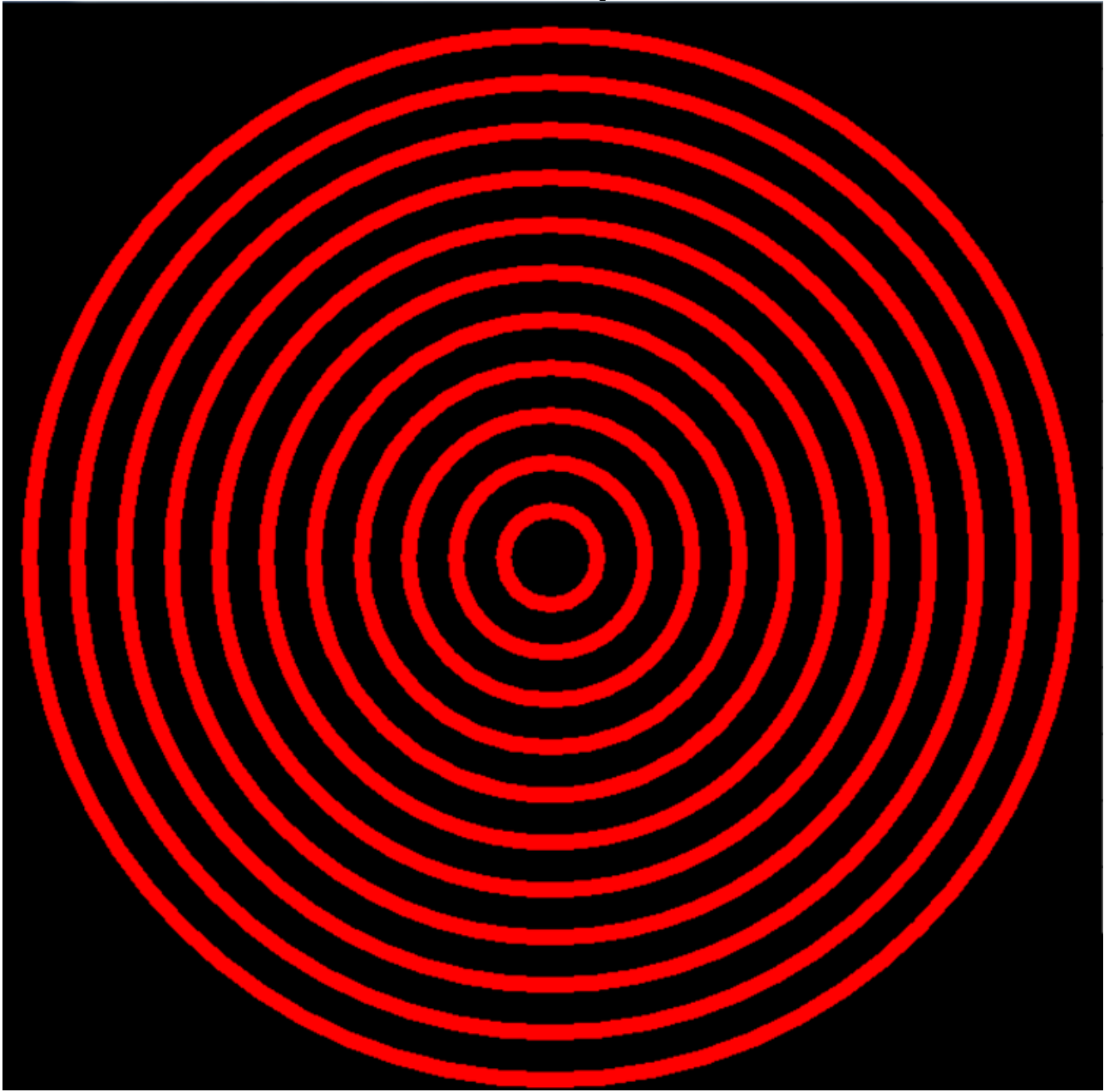
Vertical Lines

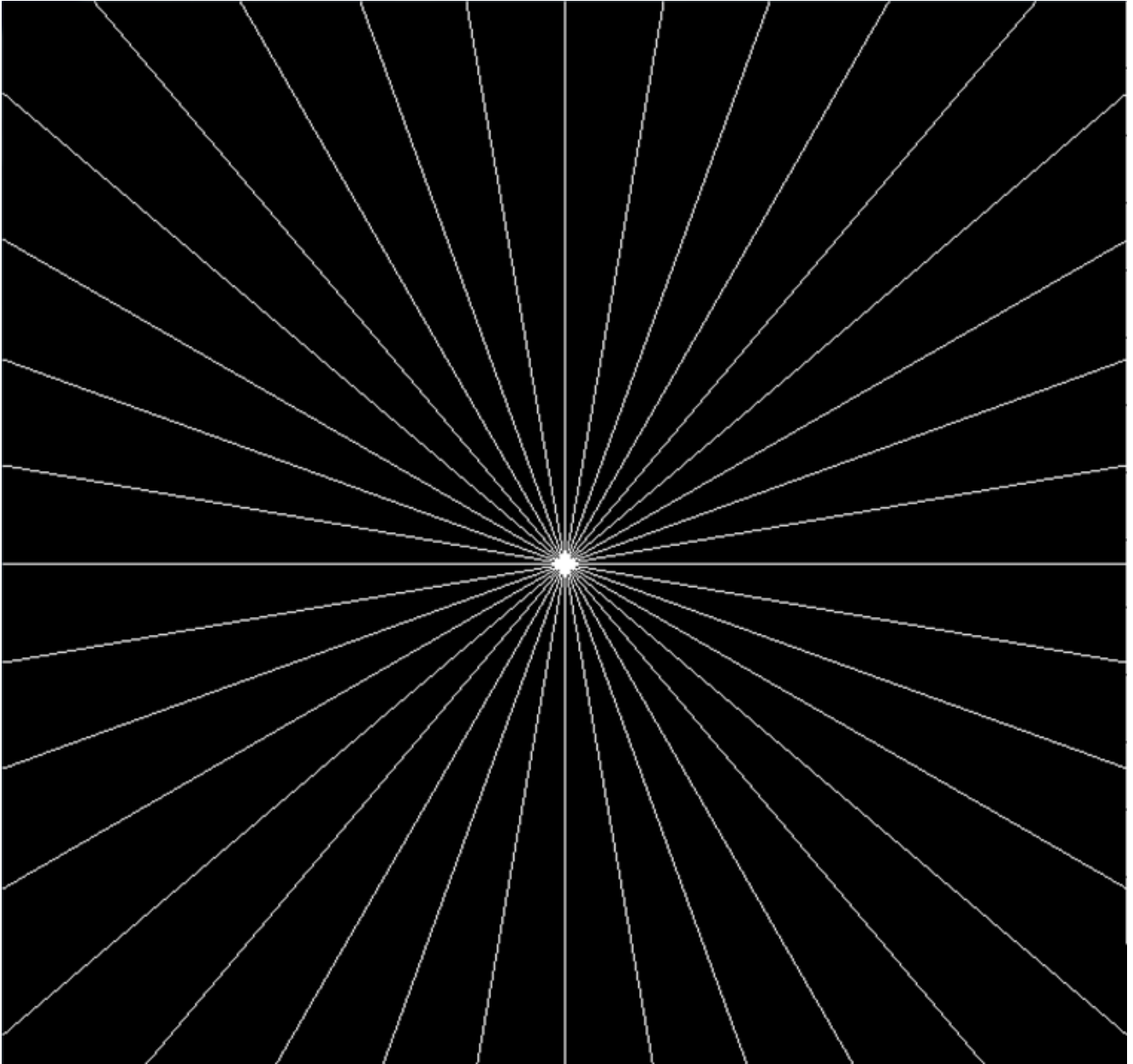Red Range

Hourglass

Line Grid

Bulls Eye

Zoom

Create a class called Looper Class that has the following methods:

Name: countUpTo10
Input: Intake feed
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into
      give followed by all the numbers up to 10. Ex: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Name: countDownFrom10
Input: Intake feed
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 10 into
      give followed by all the numbers down to 1. Ex: 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

**Fill in code for the following methods in the Myrtle Class. Remember that a Myrtle is a Turtle
so it has the following methods available:**
move()   turnRight()   turnLeft()
frontIsClear()   leftIsClear()   rightIsClear()
pickFruit()   placeFruit()   isNextToFruit()   hasFruit()

Name: move()
Input: int numSteps
Output: none
Action: Have the Myrtle move numSteps.
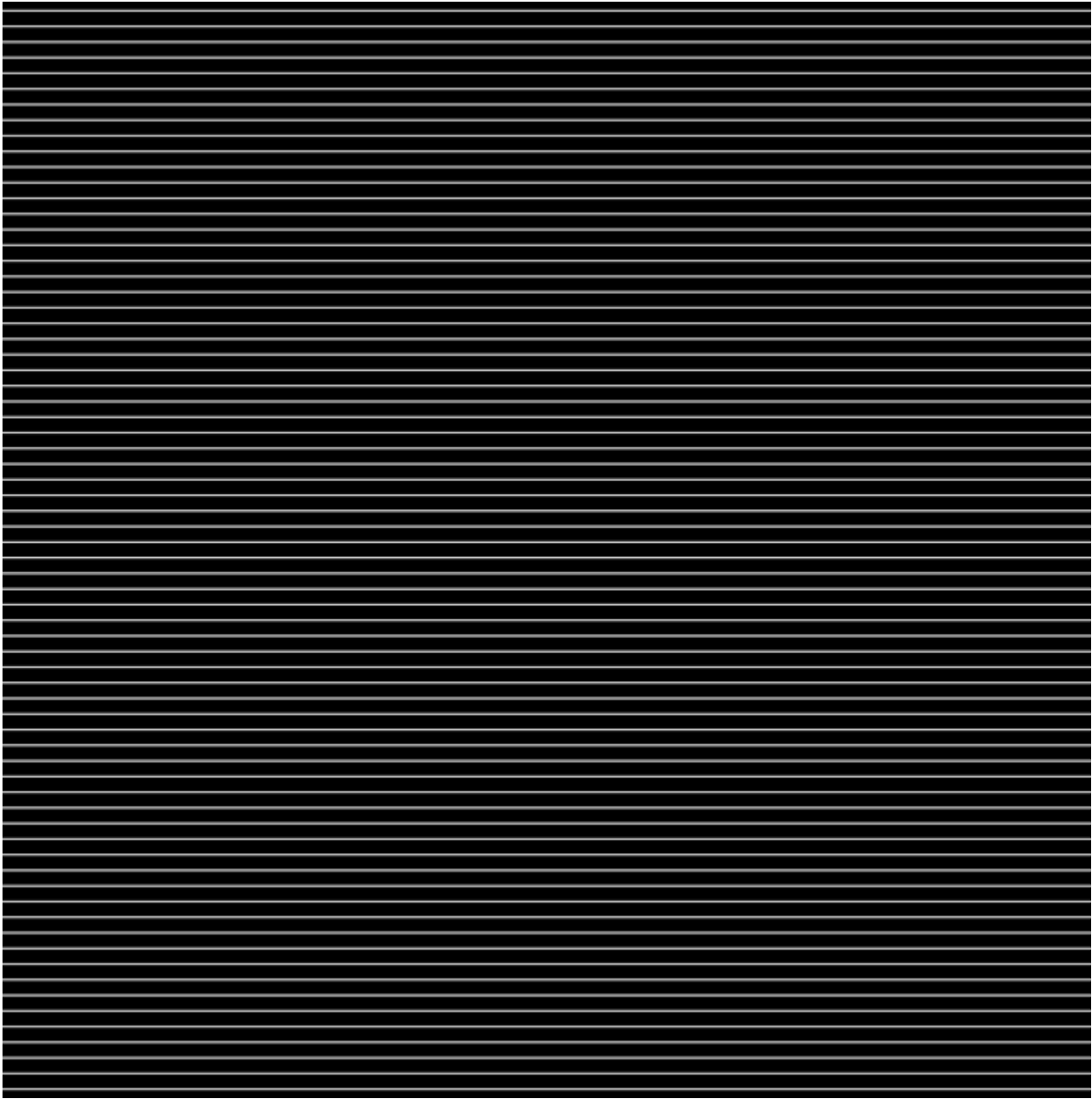
Name: moveToWall()
Input: none
Output: none
Action: Have the Myrtle move until it reaches a wall (Hint – It's front will no longer be clear if
      there is a wall infront of it)

In the Painter Class, complete the following methods to draw the graphical effect:

Horizontal Lines

Row of Boxes

Level 2

Create a class called Looper Class that has the following methods:

Name: countUpToX
Input: Intake feed, int x
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into give followed by all the numbers up to x. Ex If x is 13, then the sequence is 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

Name: countDownFromX
Input: Intake feed, int x
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number x into give followed by all the numbers down to 1. Ex If x is 13, then the sequence is 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1)

Fill in code for the following methods in the Myrtle Class. Remember that a Myrtle is a Turtle so it has the following methods available:
move()   turnRight()   turnLeft()
frontIsClear()   leftIsClear()   rightIsClear()
pickFruit()   placeFruit()   isNextToFruit()   hasFruit()
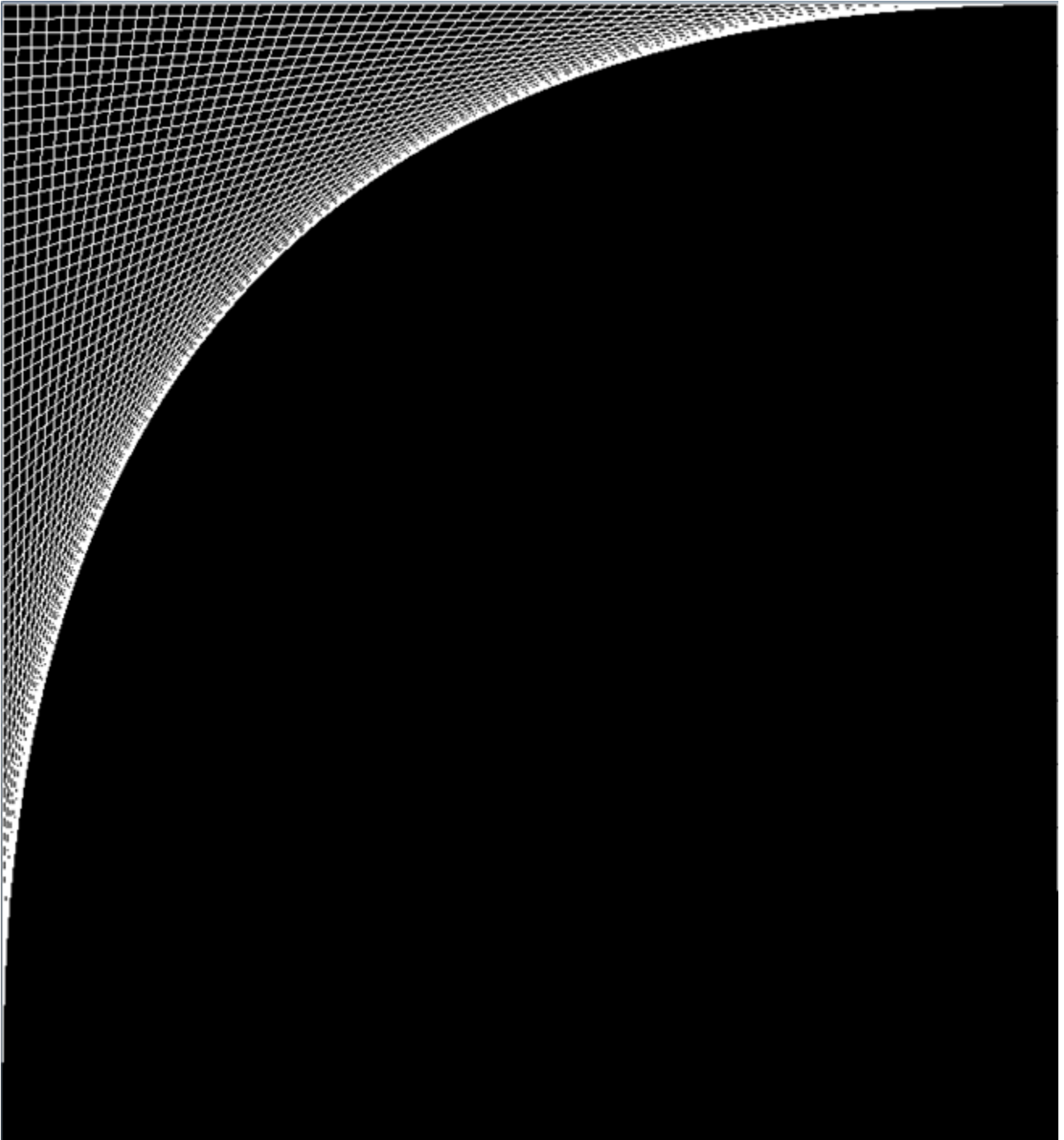
Name: moveToFruit()
Input: none
Output: none
Action: Have the Myrtle move until it reaches a fruit but stop if it runs into a wall in front of it

In the Painter Class, complete the following methods to draw the graphical effect:

Paper Stack
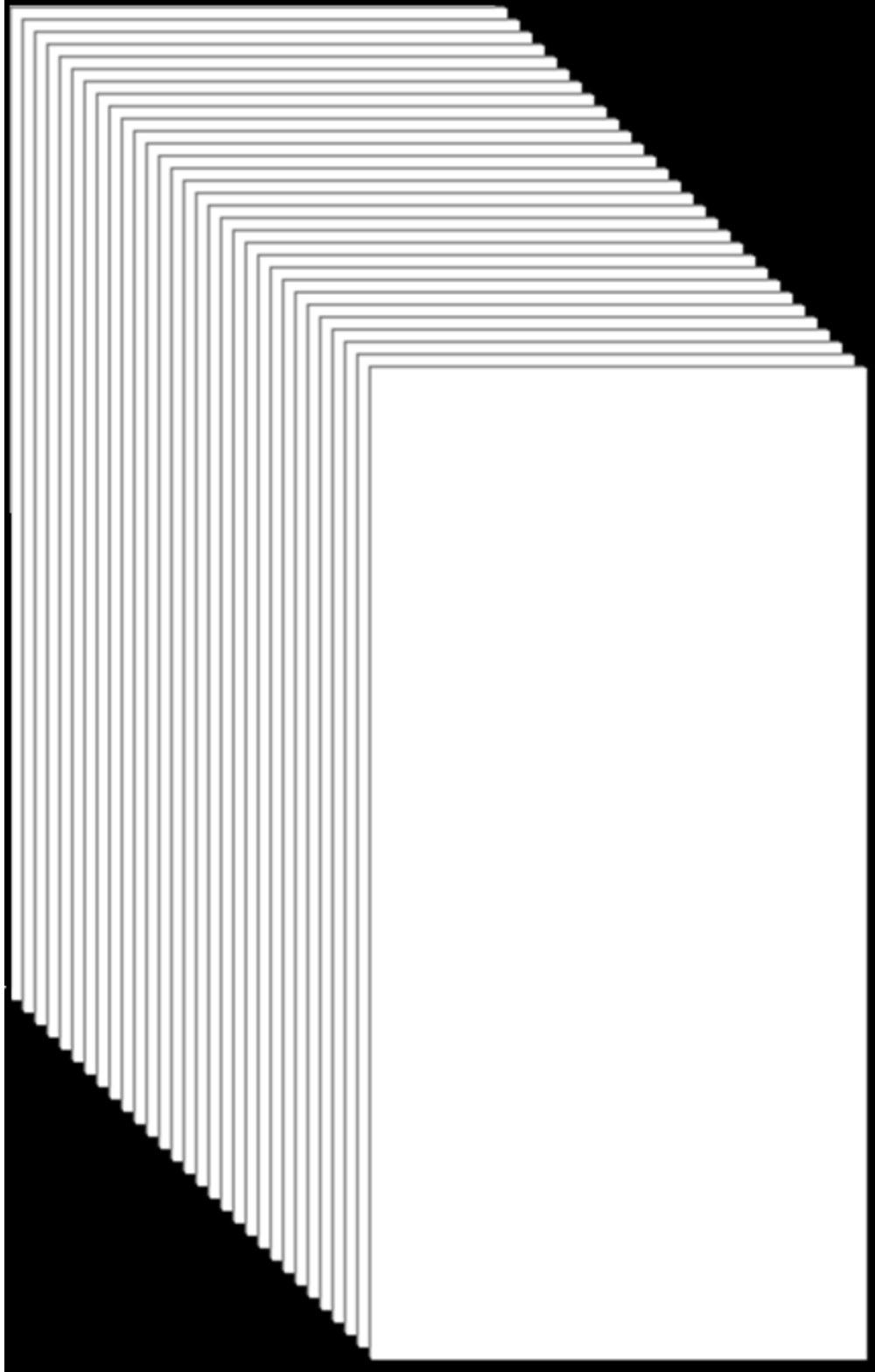
Corner Web

Create a class called Looper Class that has the following methods:

Name: countUpToXfromYbyA
Input: Intake feed, int x, int int y, int a
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number y into give followed by all the numbers up to x, going up by a each time. Ex: If x is 20 and a is 2 and y is 0, then the sequence is 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20,

Name: countDownFromXtoYbyA
Input: Intake feed, int x, int y, int a
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number x into give followed by all the numbers down to y, going down by a each time. Ex: If x is 20 and y is 0 and a is 5, then the sequence is 20, 15, 10, 5, 0

Fill in code for the following methods in the Myrtle Class. Remember that a Myrtle is a Turtle so it has the following methods available:

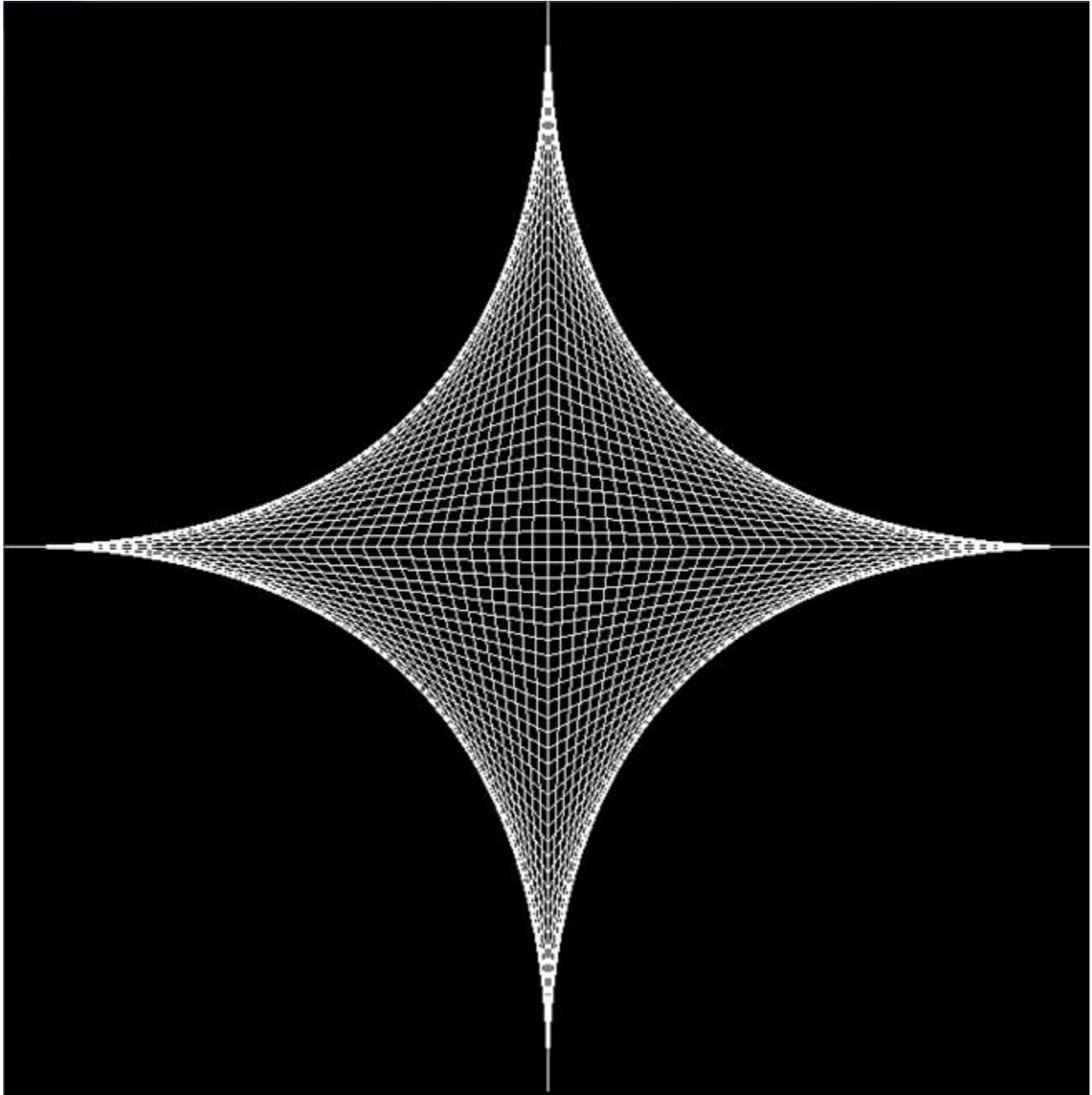|		
move()	**turnRight()**	**turnLeft()**
frontIsClear()	**leftIsClear()**	**rightIsClear()**
pickFruit()	**placeFruit()**	**isNextToFruit()** **hasFruit()**

Name: pickFruit
Input: int numFruit
Output: none
Action: Have the Myrtle pick up the numFruit indicated.

In the Painter Class, complete the following methods to draw the graphical effect:

X Box
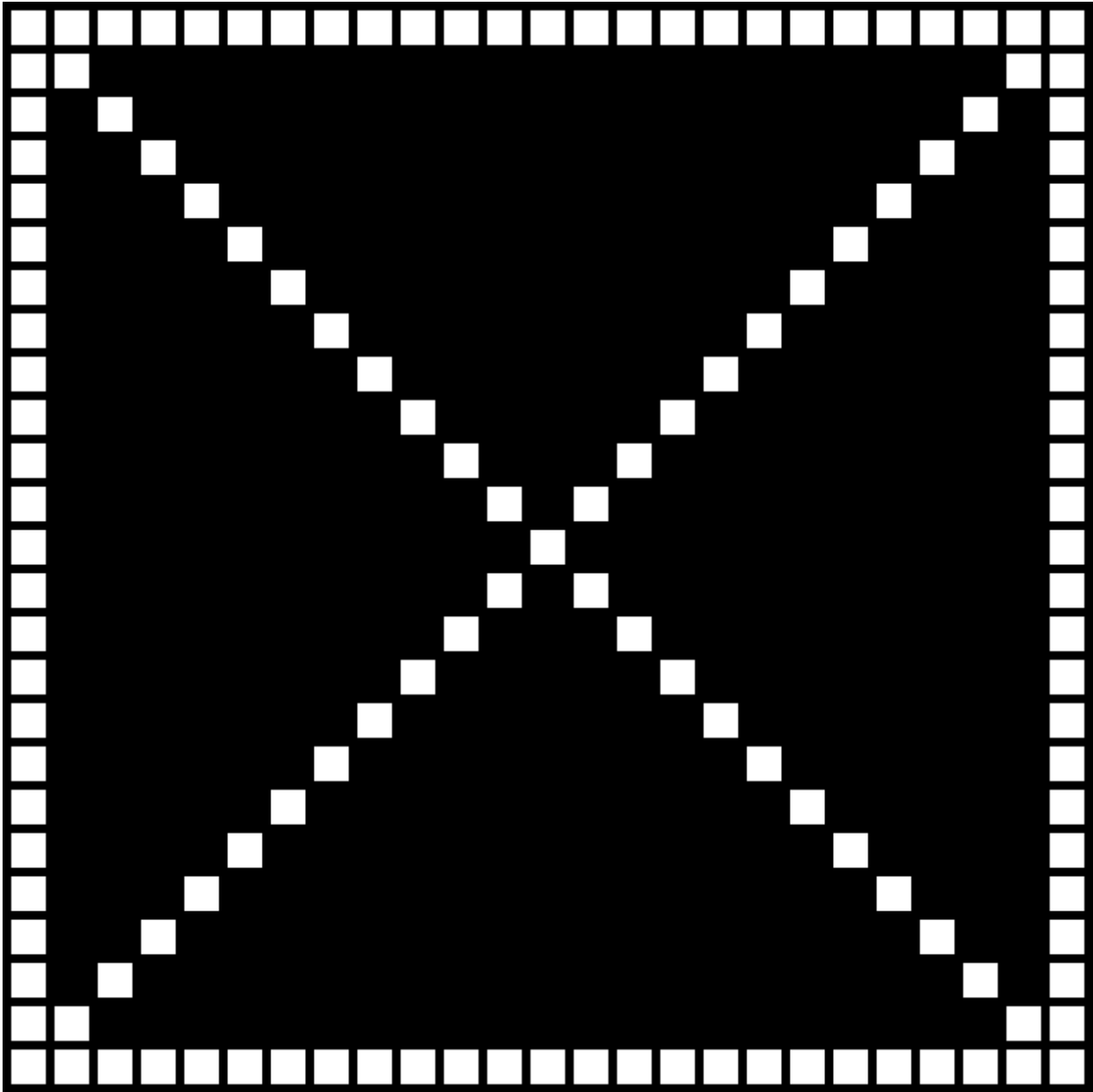
Dream Catcher

Level 4

Create a class called Looper Class that has the following methods:

Name: runSequence1
Input: Intake feed, int stop
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into
    give and then all the numbers in the following sequence until it gets to stop:
    1, 3, 7, 15, 31, 63, 127, 255, 511, …

Name: runSequence2
Input: Intake feed, int stop
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into
    give and then all the numbers in the following sequence until it gets to stop:
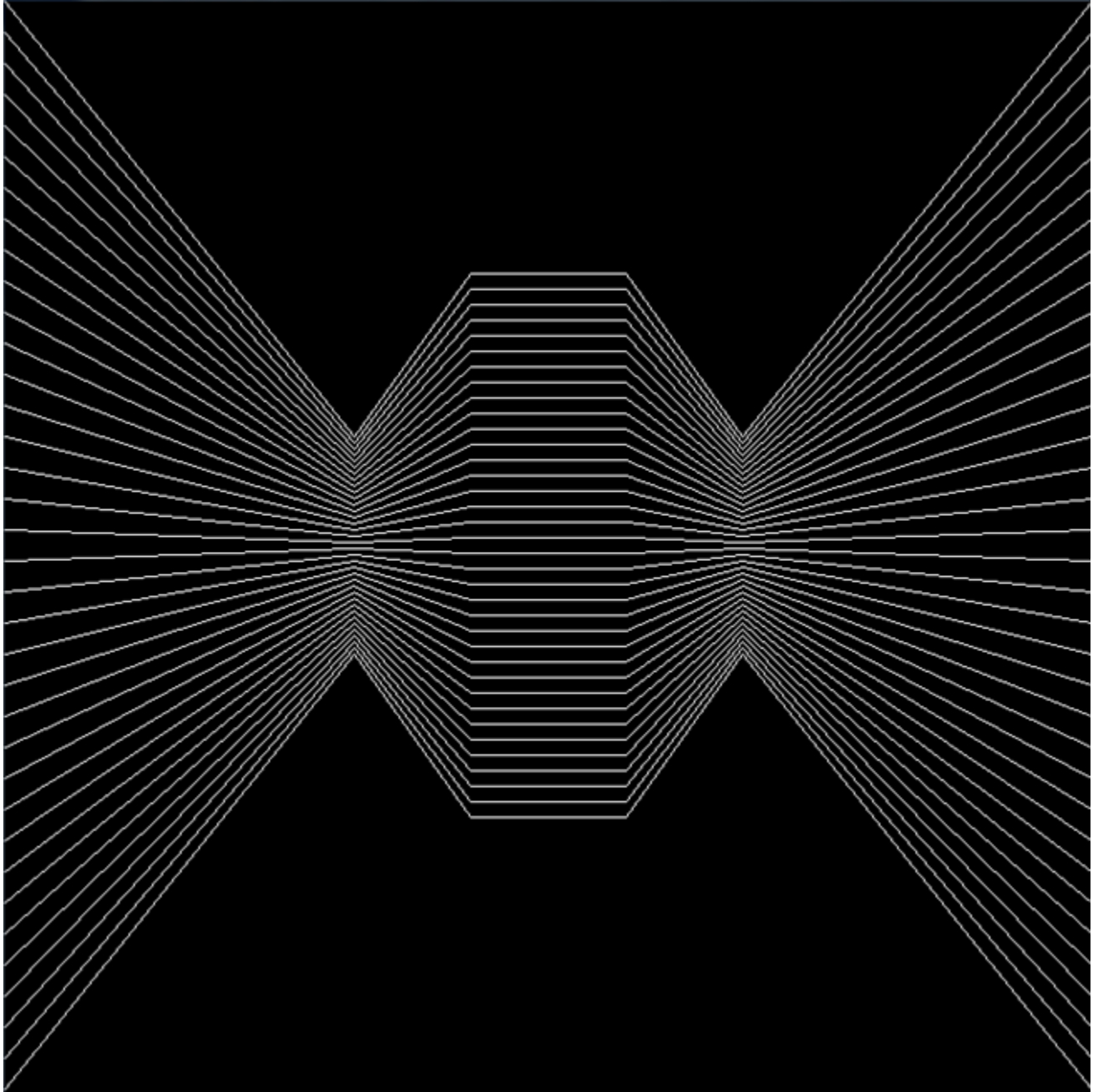   1, 4, 10, 22, 46, 94, 190, 382, 766, …

Fill in code for the following methods in the Myrtle Class. Remember that a Myrtle is a Turtle so it has the following methods available:

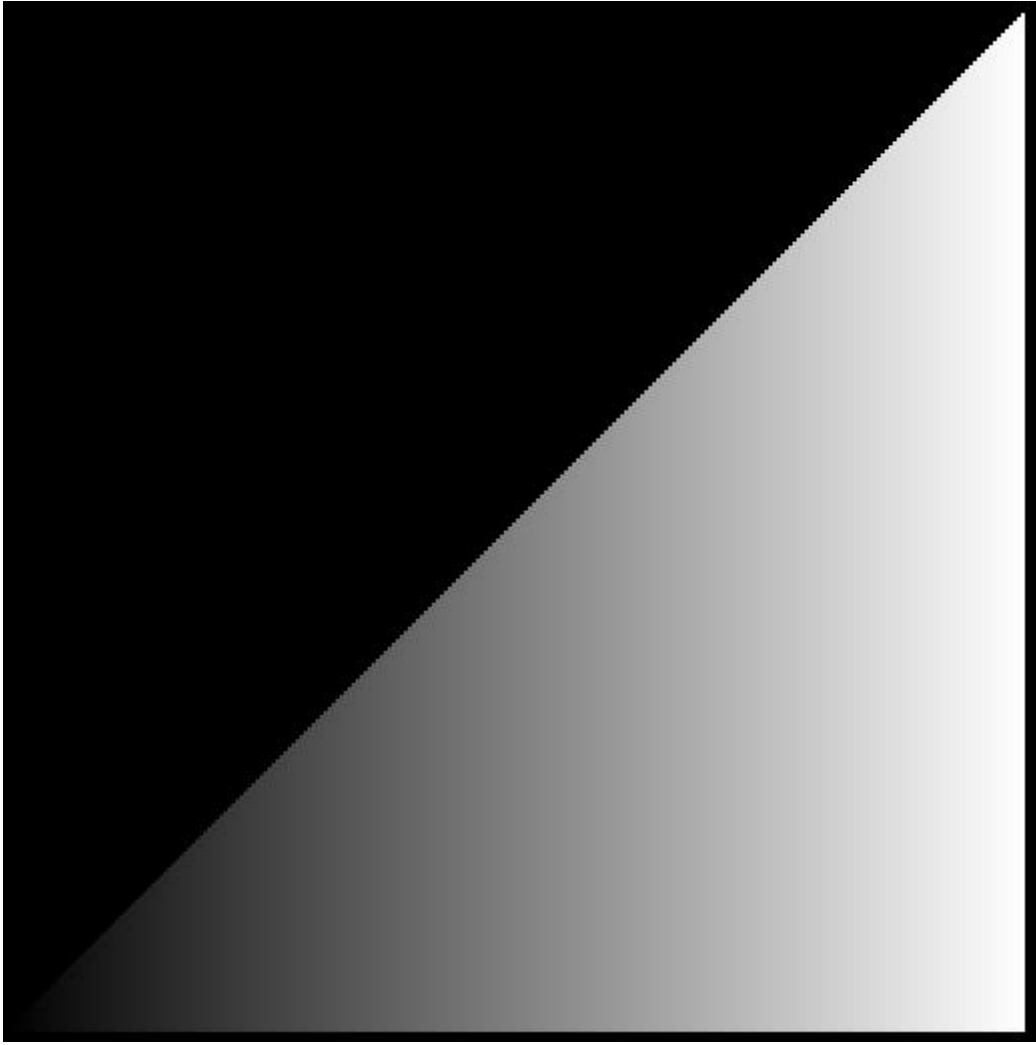|		
move()	**turnRight()**	**turnLeft()**
frontIsClear()	**leftIsClear()**	**rightIsClear()**
pickFruit()	**placeFruit()**	**isNextToFruit()** **hasFruit()**

Name: pickAllFruit
Input: none
Output: none
Action: Have the Myrtle pick up all the Fruit it is next to.

In the Painter Class, complete the following methods to draw the graphical effect:

Bow Tie

Scale

Level 5

Create a class called Looper Class that has the following methods:

Name: runSequence4
Input: Intake feed, int n
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into
    give and then alternates between -1 and 1. It does this n number of times.

Name: runSequence5
Input: Intake feed, int n
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into
    give and then all the numbers in the following sequence n number of times:
    1, -10, 100, -1000, 10000, -100000, …

**Fill in code for the following methods in the Myrtle Class. Remember that a Myrtle is a Turtle
so it has the following methods available:**
move()   turnRight()   turnLeft()
frontIsClear()   leftIsClear()   rightIsClear()
pickFruit()   placeFruit()   isNextToFruit()   hasFruit()
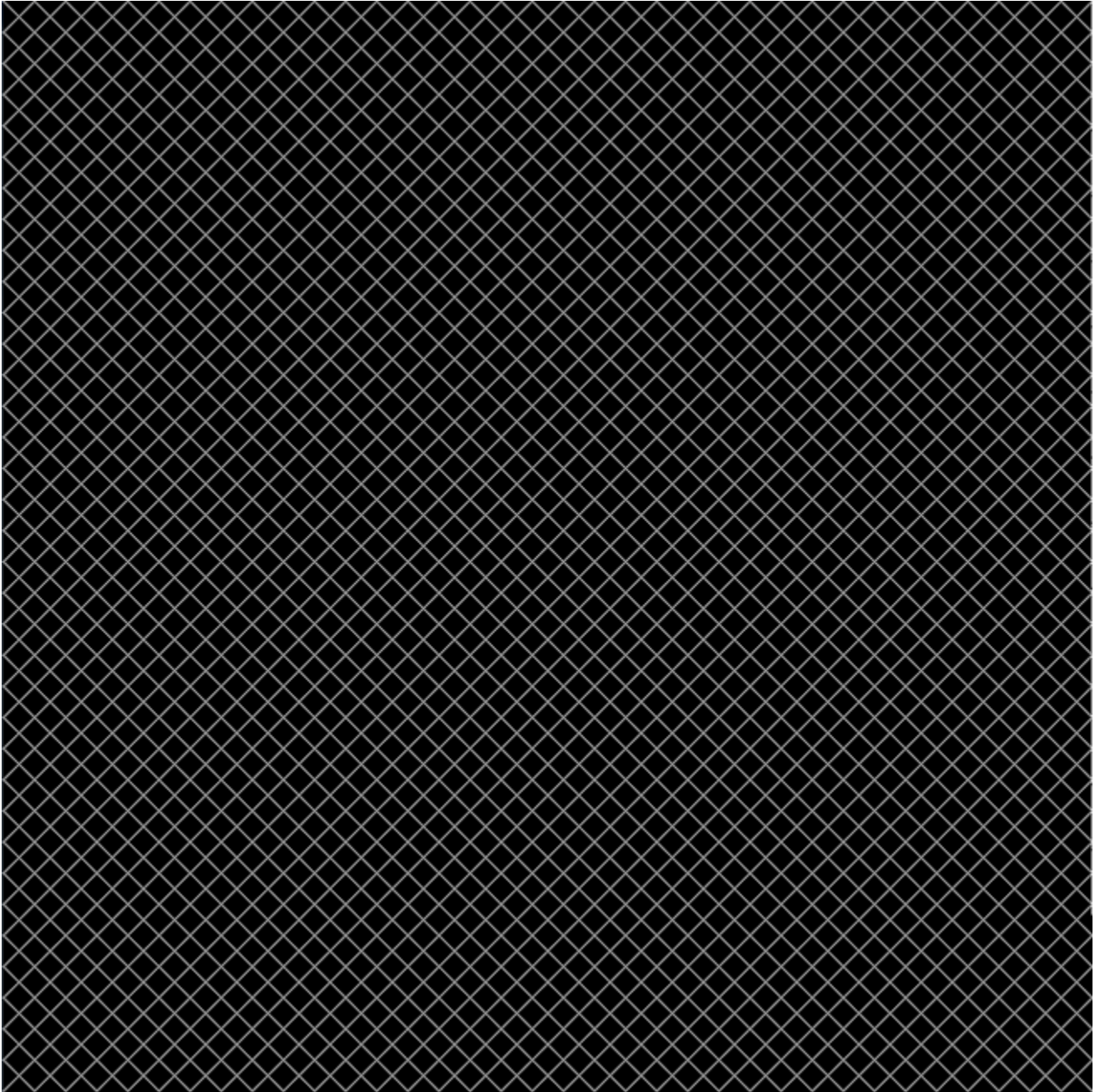
Name: placeRow
Input: int numSteps
Output: none
Action: Have the Myrtle move numSteps and place one fruit at each location it moves to.

In the Painter Class, complete the following methods to draw the graphical effect:

Rainbow Road

Criss Cross

Level 6

Create a class called Looper Class that has the following methods:

Name: runSequence7
Input: Intake feed, int stop
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into
     give and then all the numbers in the following sequence until it gets to stop:
     1, 2, 4, 7, 11, 14, 16, 17, 17, 18, 20, 23, 27, …

Name: runSequence8
Input: Intake feed, int n
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into
     give and then all the numbers in the following sequence until it has output n numbers:
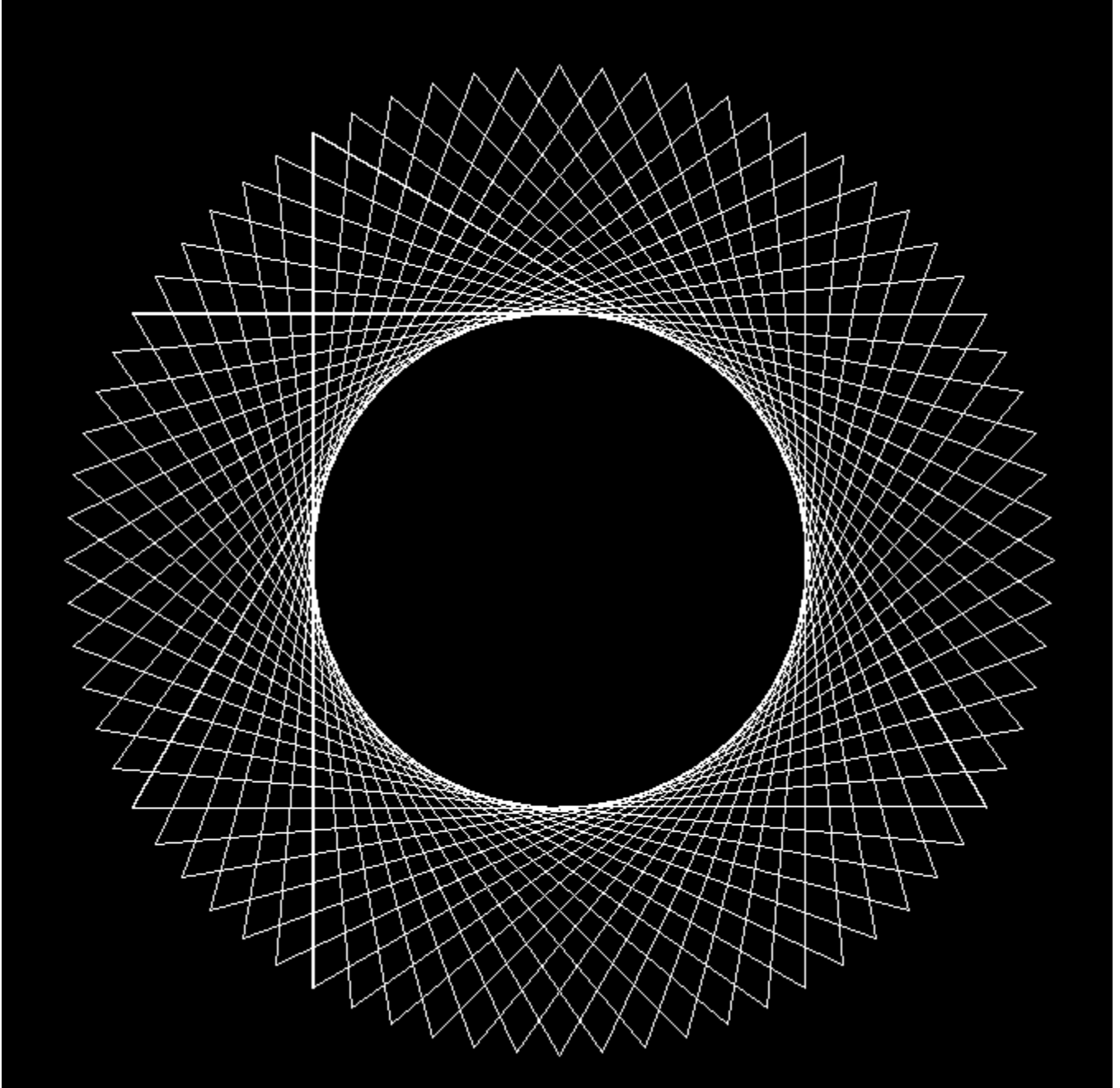   1, 2, 2, 3, 3, 3, 4, 4, 4, 4, …

Fill in code for the following methods in the Myrtle Class. Remember that a Myrtle is a Turtle so it has the following methods available:

move()   **turnRight()**   **turnLeft()**
frontIsClear()   **leftIsClear()**   **rightIsClear()**
pickFruit()   **placeFruit()**   **isNextToFruit()**   **hasFruit()**

Name: findFruitInMaze
Input: none
Output: none
Action: Have the Myrtle find the fruit in the maze it was placed in.

In the Painter Class, complete the following methods to draw the graphical effect:

Hole

Spiral