

COW – Nested Looping

Level 1

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

Name: countUp
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 the numbers 1 to 10. It then passes in a 2 followed by the numbers 1 to 10. It keeps doing the same except for adding to the first number by 1. It does this until the first number reaches 10 and the numbers 1 to 10 are passed in. Ex:

```
1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
2, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
3, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
4, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
6, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
7, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
8, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

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:

Movement Methods: move(), turnRight(), turnLeft()

Path Methods: frontIsClear(), leftIsClear(), rightIsClear()

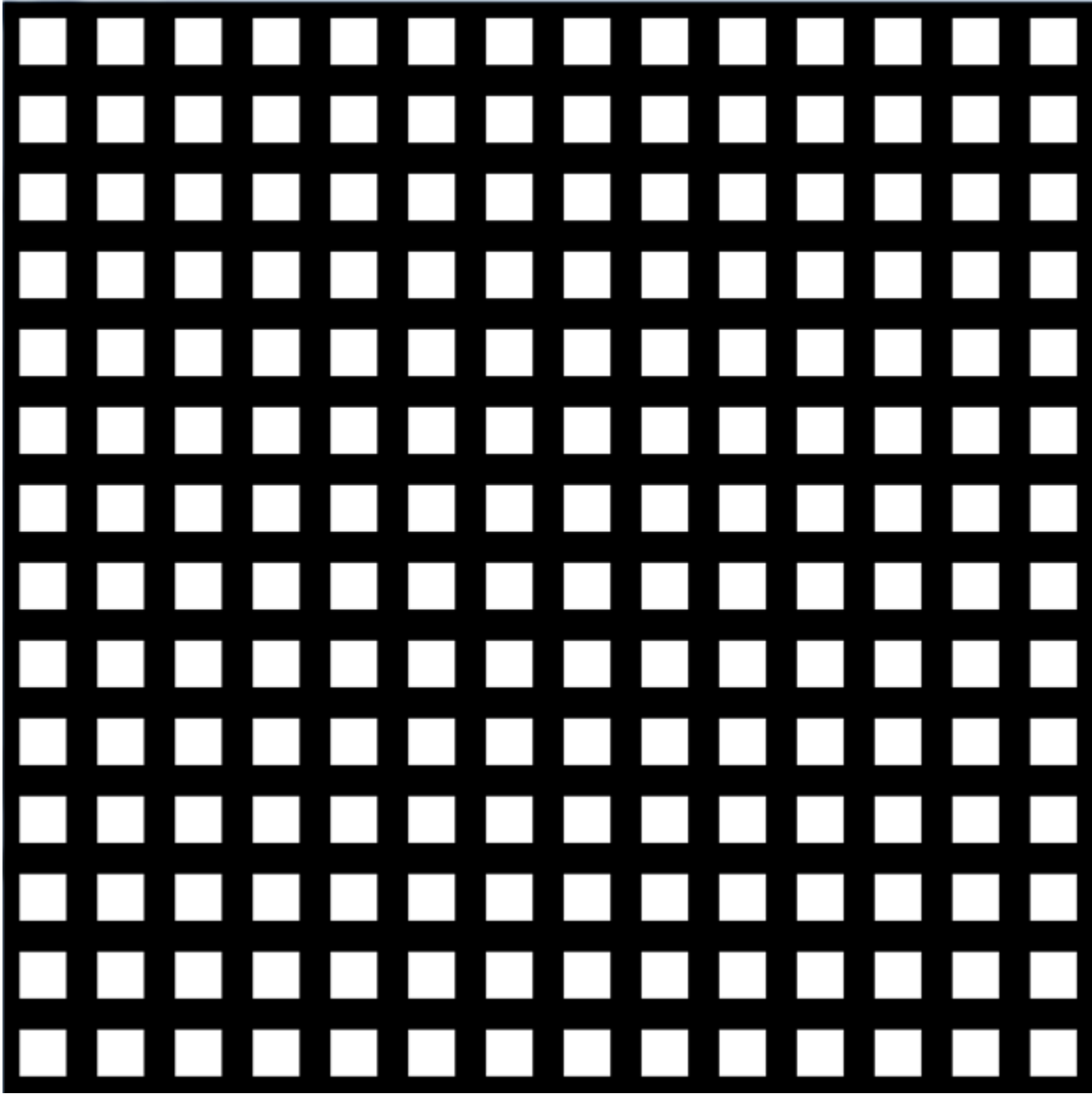
Fruit Methods: pickApple(), pickOrange(), pickLemon(),
placeApple(), placeOrange(), placeLemon(),
isNextToApple(), isNextToOrange(), isNextToLemon(),
hasApple(), hasOrange(), hasLemon()

Name: pickRowOfApples
Input: int numSteps
Output: none
Action: Have the Myrtle move numSteps and pick up all the apples at each location it moves to.

Name: pickApplesToWall(int numApples)
Input: none
Output: none
Action: Have the Myrtle move until it hits a wall and pick up numApples at each location it moves to.

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

Box Grid



Level 2

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

Name: countUp
Input: Intake feed, int numRows, int numCols
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into give followed by the numbers 1 to numCols. It then passes in a 2 followed by the numbers 1 to numCols. It keeps doing the same except for adding to the first number by 1. It does this until the first number reaches numRows and the numbers 1 to numCols are passed in.
Ex (numCols – 4, numCols – 7):

1, 1, 2, 3, 4, 5, 6, 7
2, 1, 2, 3, 4, 5, 6, 7
3, 1, 2, 3, 4, 5, 6, 7
4, 1, 2, 3, 4, 5, 6, 7

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:

Movement Methods: move(), turnRight(), turnLeft()

Path Methods: frontIsClear(), leftIsClear(), rightIsClear()

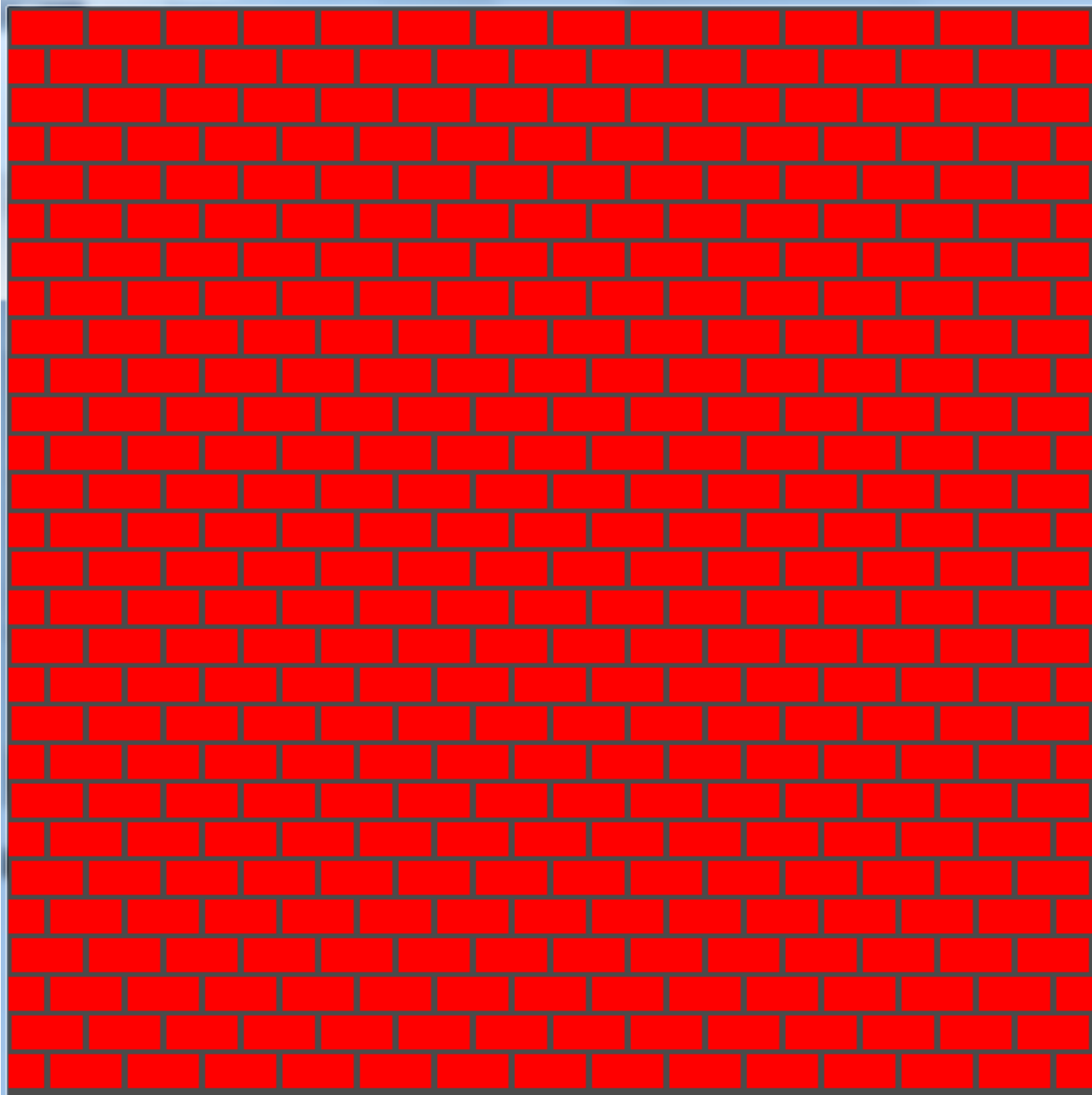
Fruit Methods: pickApple(), pickOrange(), pickLemon(),
placeApple(), placeOrange(), placeLemon(),
isNextToApple(), isNextToOrange(), isNextToLemon(),
hasApple(), hasOrange(), hasLemon()

Name: pickRowOfFruit
Input: int numSteps
Output: none
Action: Have the Myrtle move numSteps and pick up all the fruit at each location it moves to.

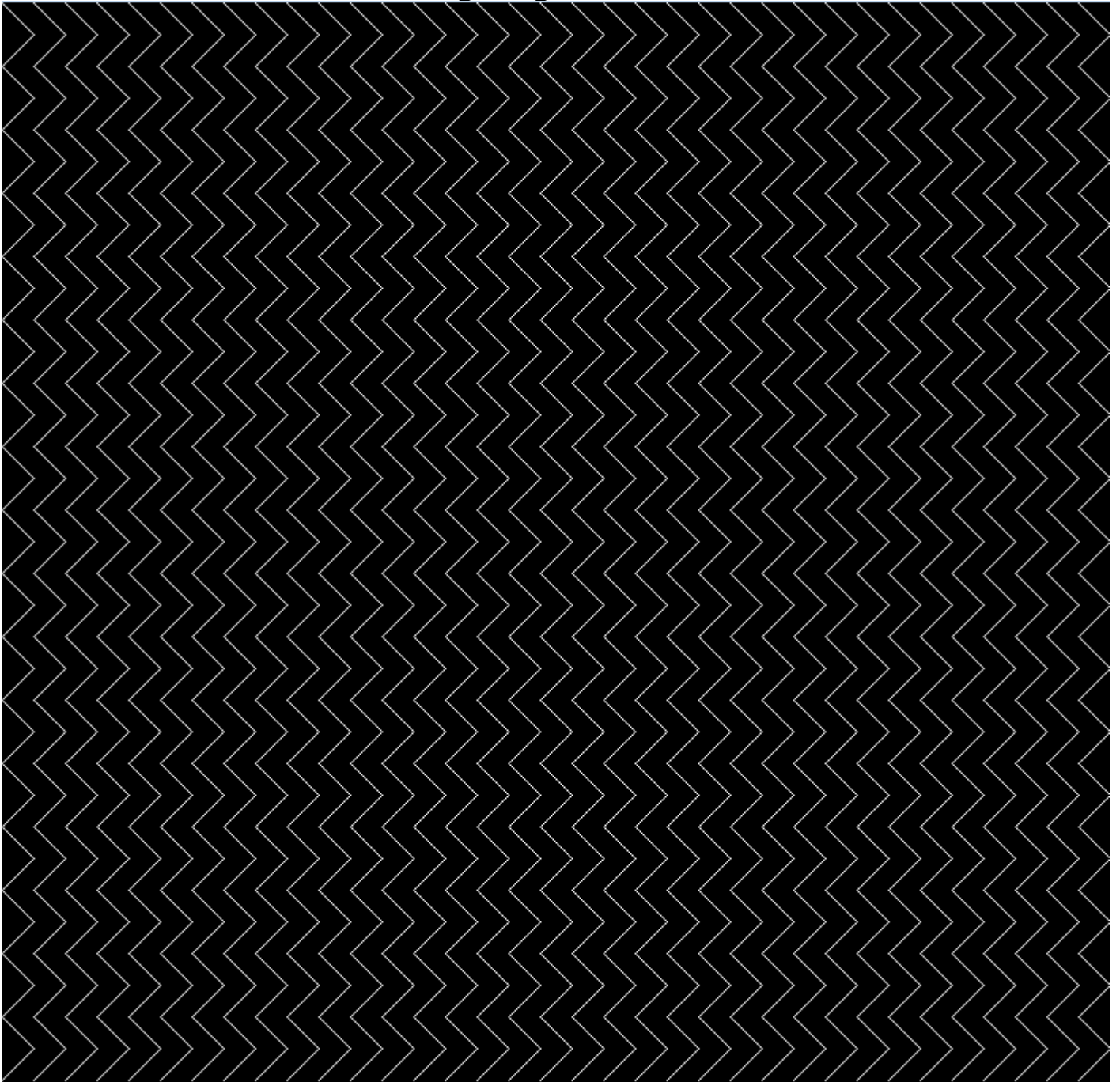
Name: pickFruitToWall(int numApples, int numOranges, int numLemons)
Input: none
Output: none
Action: Have the Myrtle move until it hits a wall and pick up numApples, numOranges, and numLemons at each location it moves to.

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

Brick Wall



Zig Zag (30)



Level 3

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

Name: countUp
Input: Intake feed, int numRows
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into give followed by the numbers 1 up to current row. It then passes in a 2 followed by the numbers 1 up to current row. It keeps doing the same except for adding to the first number by 1. It does this until the first number reaches numRows and the numbers 1 up to current row are passed in. Ex (numRows – 4):

1, 1,
2, 1, 2,
3, 1, 2, 3,
4, 1, 2, 3, 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:

Movement Methods: move(), turnRight(), turnLeft()

Path Methods: frontIsClear(), leftIsClear(), rightIsClear()

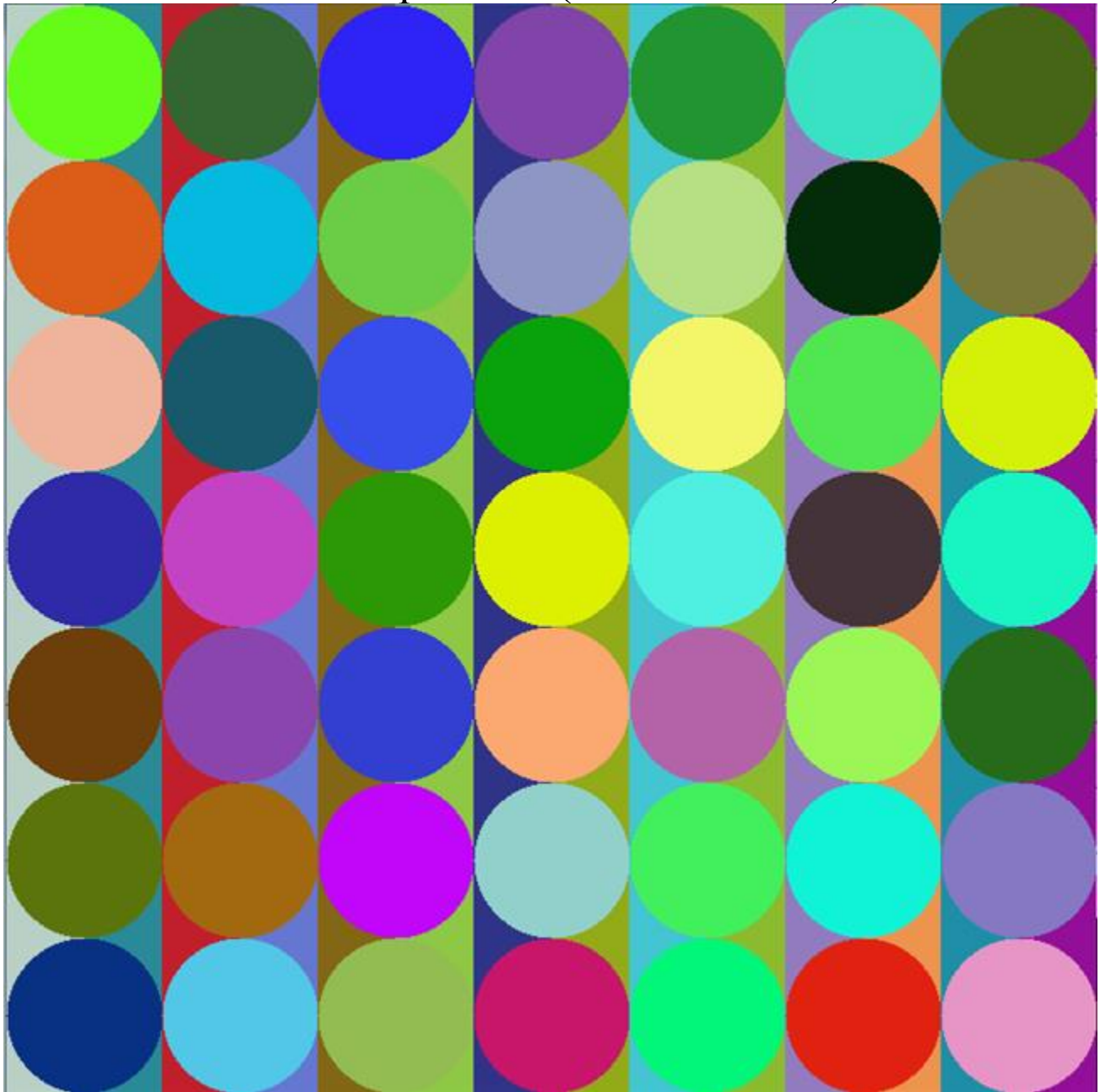
Fruit Methods: pickApple(), pickOrange(), pickLemon(),
placeApple(), placeOrange(), placeLemon(),
isNextToApple(), isNextToOrange(), isNextToLemon(),
hasApple(), hasOrange(), hasLemon()

Name: harvestRowAndDeposit
Input: int length
Output: none
Action: Have the Myrtle harvest all the fruit in a row of given length and then deposit all the fruit at the location that Myrtle started. Also have Myrtle face the same direction that Myrtle was facing at the start. Assume that the location of the first fruit is in the location in front of Myrtle and not where Myrtle is standing.

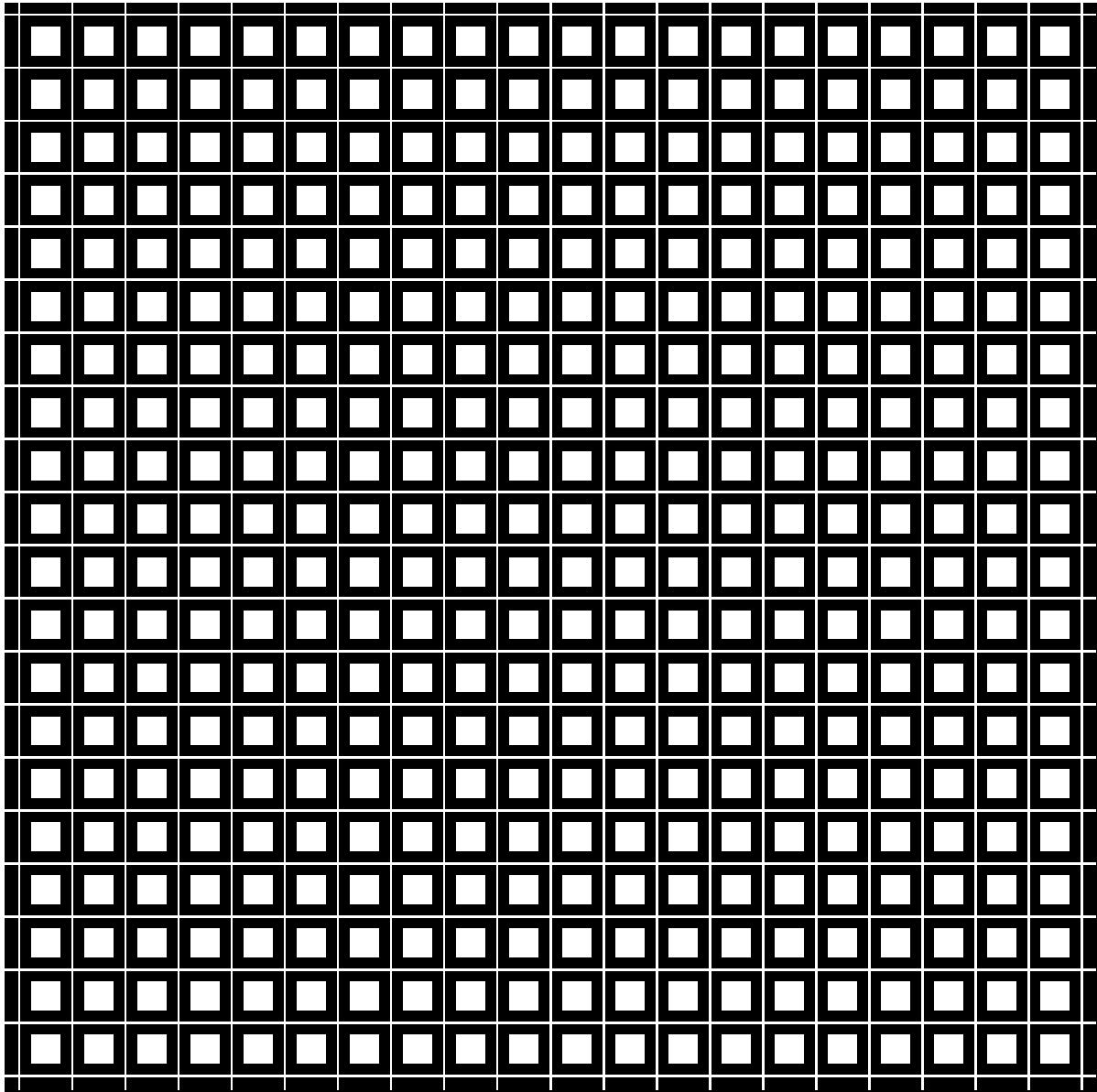
Name: placeRowOfFruit
Input: int length, int numApples, int numOranges, int numLemons
Output: none
Action: Have the Myrtle place a row of fruit of given length with the number of apples, oranges, and lemons indicated at each location. Then have Myrtle move back to where Myrtle started and face the same direction. Assume that the location of the first fruit is in the location in front of Myrtle and not where Myrtle is standing.

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

Wall Paper 70's (random colors)



Grid & Lines



Level 4

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

Name: countUpByRow
Input: Intake feed, int numRows, int numValues
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into give followed a certain number of values that start at 1 and increasing by the row number from one value to the other. It then passes in a 2 followed a certain number of values that start at 1 and increasing by the row number from one value to the other. It does this until the first number reaches numRows.
Ex (numRows – 4, numValues – 6):

1, 1, 2, 3, 4, 5, 6
2, 1, 3, 5, 7, 9, 11
3, 1, 4, 7, 10, 13, 16
4, 1, 5, 9, 13, 17, 21

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:

Movement Methods: move(), turnRight(), turnLeft()

Path Methods: frontIsClear(), leftIsClear(), rightIsClear()

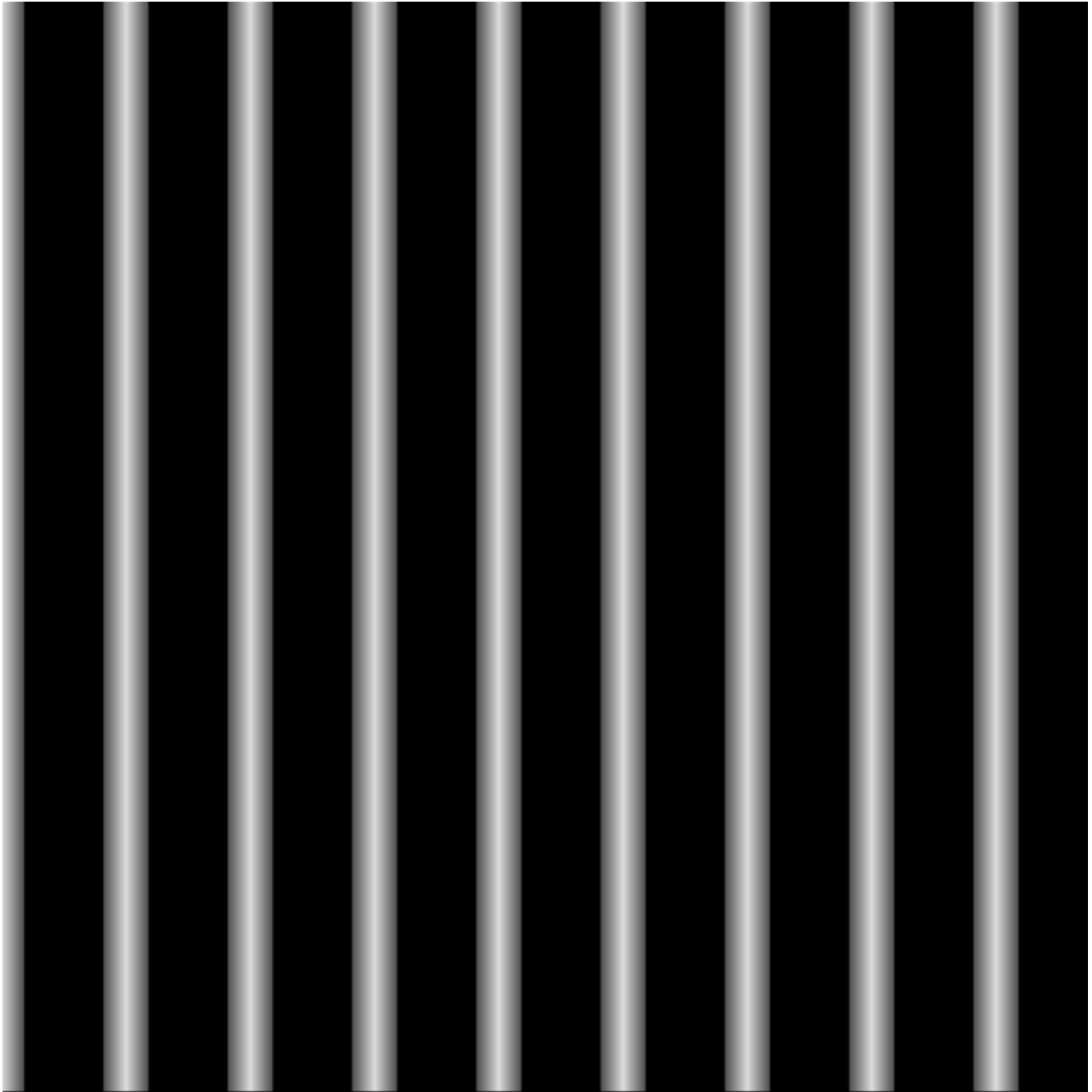
Fruit Methods: pickApple(), pickOrange(), pickLemon(),
placeApple(), placeOrange(), placeLemon(),
isNextToApple(), isNextToOrange(), isNextToLemon(),
hasApple(), hasOrange(), hasLemon()

Name: harvestAllAndDeposit
Input: int width, int height
Output: none
Action: Have the Myrtle harvest a grid of fruit with the indicated width and height. Assume that the Myrtle starts in the lower left hand corner of where the grid it is supposed to be. Also assume that there could be multiple fruits at each location. All fruits harvested in each row should be placed at the beginning of each row. Myrtle should end up at the beginning of the top row.

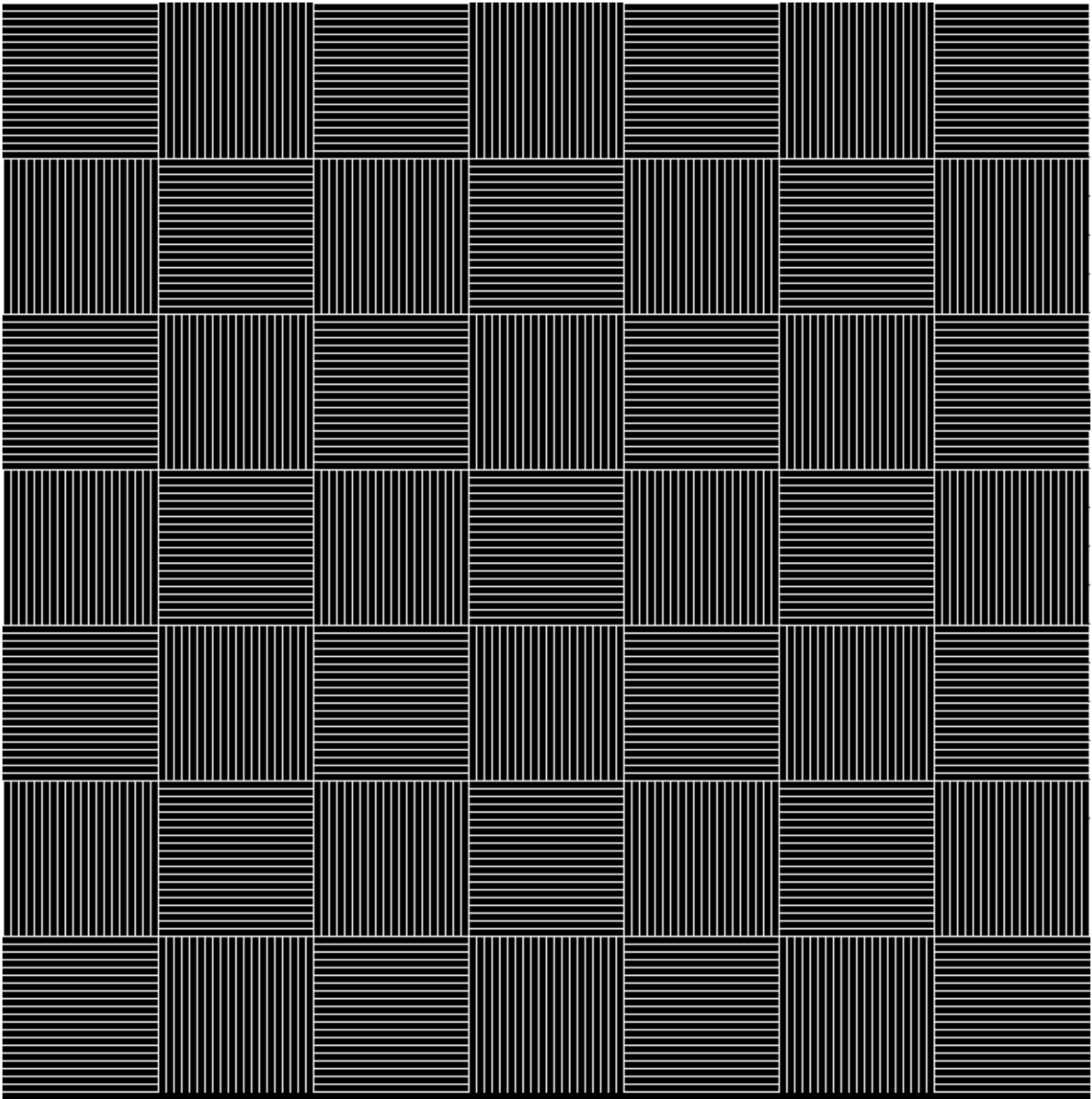
Name: placeGrid
Input: int width, int height, int numApples, int numOranges, int numLemons
Output: none
Action: Have the Myrtle place a grid of fruit with the indicated width and height. Assume that the Myrtle starts in the lower left hand corner of the grid it is supposed to place. The number of fruit at each location should be equal to numApples, numOranges, numLemons. Myrtle should end up at the beginning of the top row.

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

Tubular Two



Patchwork Max



Level 5

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

Name: countUpToCeilingAndBack

Input: Intake feed, int numRows, int ceiling

Output: none

Action: Takes in an intake object and calls the method give(). It first passes the number 1 into give followed a certain number of values that start at 1 and increasing by the row number from one value to the other until it reaches or exceeds ceiling. It then counts back down to 1. It then passes in a 2 followed a certain number of values that start at 1 and increasing by the row number from one value to the other until it reaches or exceeds ceiling. It then counts back down to 1. It does this until the first number reaches numRows.

Ex (numRows – 4, ceiling – 15):

```
1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
2, 1, 3, 5, 7, 9, 11, 13, 15, 13, 11, 9, 7, 5, 3, 1
3, 1, 4, 7, 10, 13, 16, 13, 10, 7, 4, 1
4, 1, 5, 9, 13, 17, 13, 9, 5, 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:

Movement Methods: move(), turnRight(), turnLeft()

Path Methods: frontIsClear(), leftIsClear(), rightIsClear()

Fruit Methods: pickApple(), pickOrange(), pickLemon(),
placeApple(), placeOrange(), placeLemon(),
isNextToApple(), isNextToOrange(), isNextToLemon(),
hasApple(), hasOrange(), hasLemon()

Name: searchForFruit

Input: none

Output: none

Action: Have the Myrtle keep moving until it moves next to the fruit. You may assume that Myrtle starts in the lower left hand corner and is facing EAST. You may assume there will be one apple, one orange, and one lemon at that one location.

Name: placeSpiral

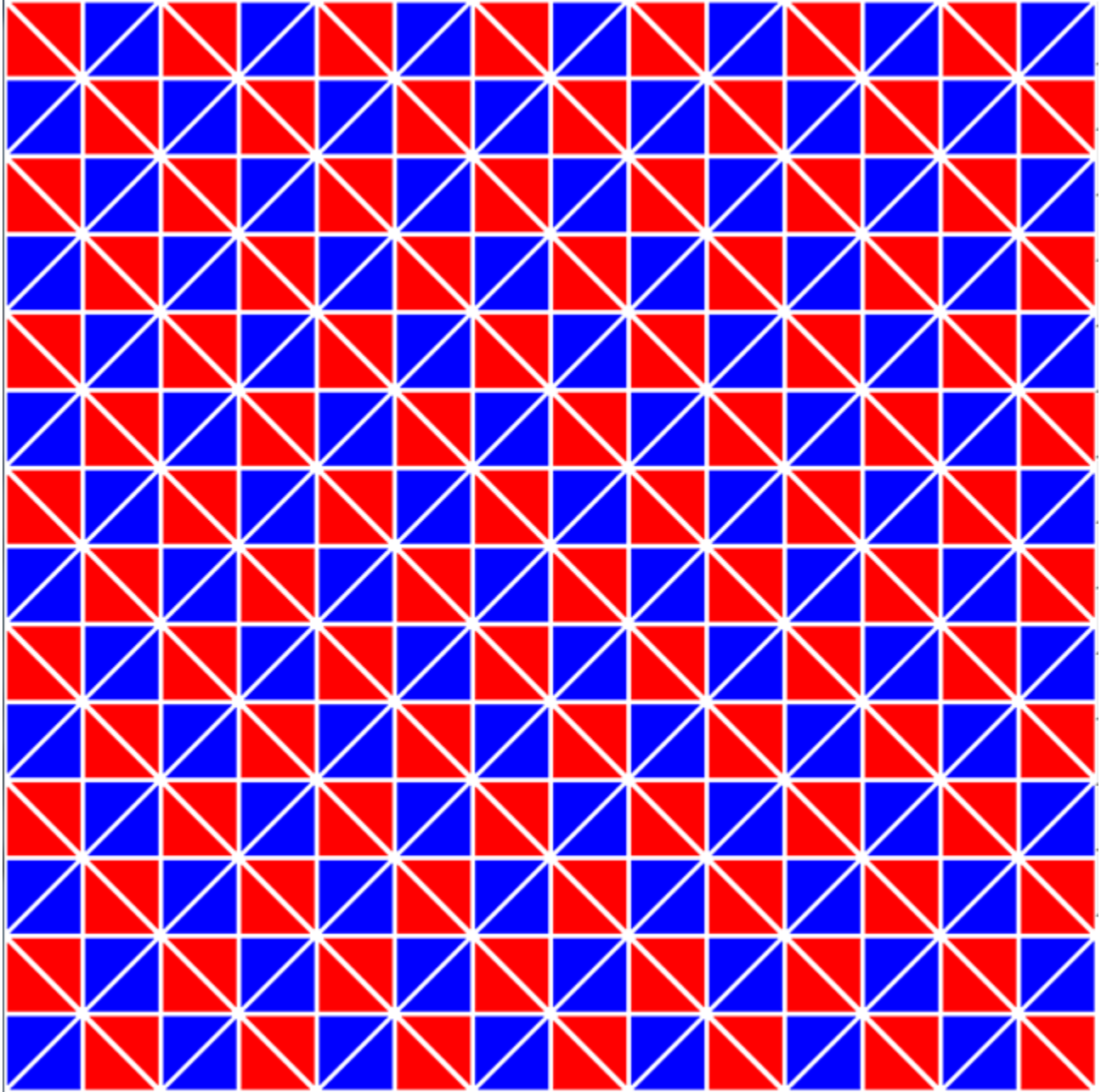
Input: int numSides

Output: none

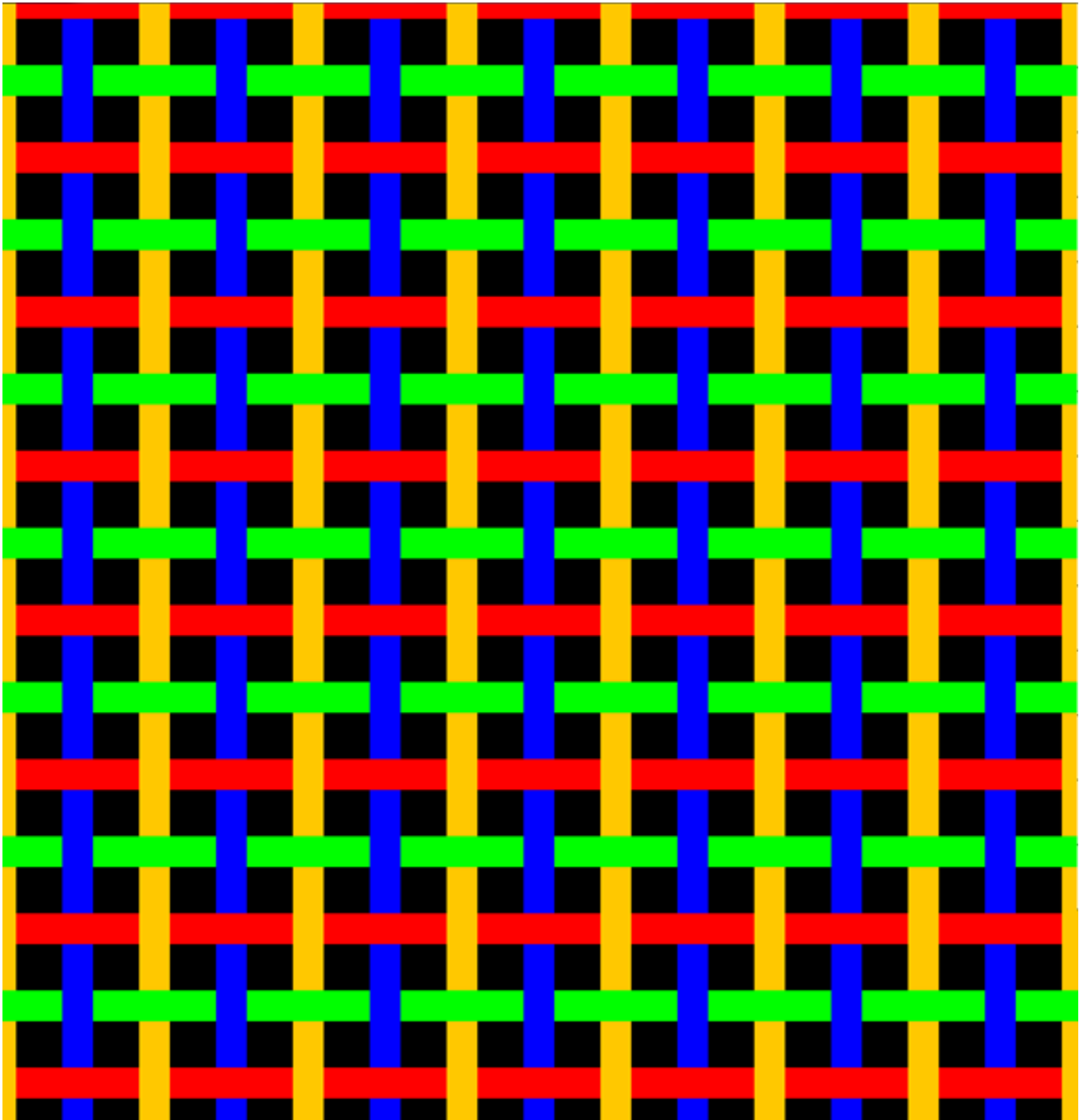
Action: Have Myrtle place fruit in a spiral with the number of sides indicated. Each box should have one apple, one orange, and one lemon placed in it. The spiral should have a spacing of one block and start by going east and spiral counter clockwise. You may assume that Myrtle starts in the center facing east and you do not need to worry about Myrtle running into a wall.

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

Tiled Floor



Patchwork



Level 6

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

Name: fluctuateToCeiling
Input: Intake feed, int numRows
Output: none
Action: Takes in an intake object and calls the method give(). It first passes the number 1 into give followed by zero up to the numRows number followed by the values down to the negative value of the row followed by values up to zero. It then add one to the row number and repeats the process. It does this until the first number reaches numRows.
Ex (numRows = 4):

```
1, 0, 1, 0, -1, 0
2, 0, 1, 2, 1, 0, -1, -2, -1, 0
3, 0, 1, 2, 3, 2, 1, 0, -1, -2, -3, -2, -1, 0
4, 0, 1, 2, 3, 4, 3, 2, 1, 0, -1, -2, -3, -4, -3, -2, -1, 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:

Movement Methods: move(), turnRight(), turnLeft()

Path Methods: frontIsClear(), leftIsClear(), rightIsClear()

Fruit Methods: pickApple(), pickOrange(), pickLemon(),
placeApple(), placeOrange(), placeLemon(),
isNextToApple(), isNextToOrange(), isNextToLemon(),
hasApple(), hasOrange(), hasLemon()

Name: placePyramid
Input: int sideLength
Output: none
Action: Have the Myrtle place a grid of fruits with a sideLength equal to sideLength. You may assume that Myrtle starts in the lower left hand corner and is facing EAST. The outside edge of the grid should have one fruit each. Each successive location toward the center will have one more fruit then the one further out. Ex (sideLength = 5)

```
1 1 1 1 1
1 2 2 2 1
1 2 3 2 1
1 2 2 2 1
1 1 1 1 1
```

Name: fillMazeWithFruits

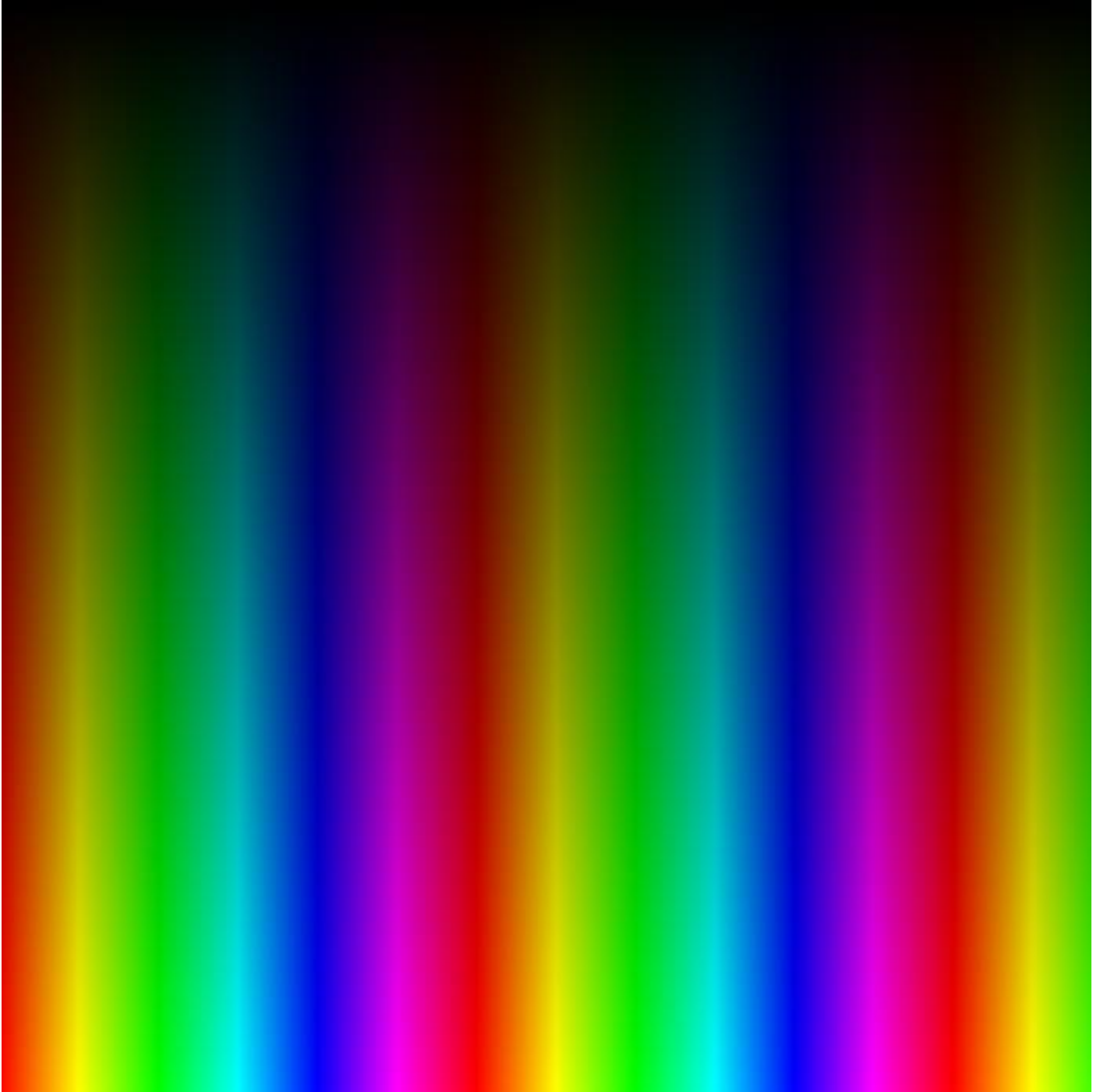
Input: none

Output: none

Action: Have the Myrtle place one of each fruit at each location in a maze. You can assume that Myrtle starts at a dead end facing in.

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

Rainbow Fade



Disco Down

