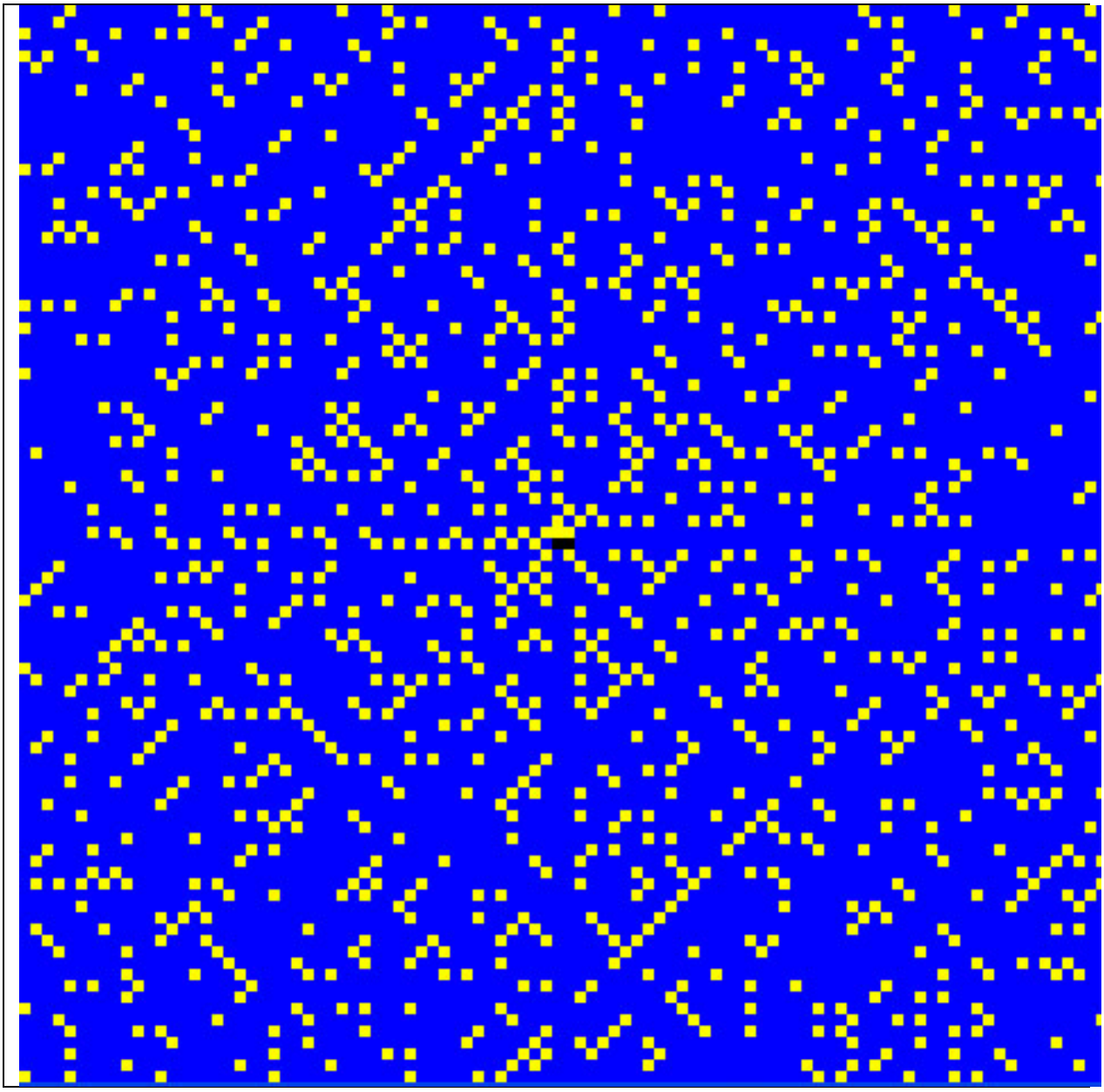
Physics 5 – DarkGDK003: Spiraling Primes

In this project you will write code to produce the rectangular spiral described in the video at http://vihart.com/doodling/ - the last one, titled Doodling in Math: Sick Number Games.

The goal is to color a sequence of dbBox either one color or another depending on whether its index is prime or composite, and to arrange these dbBox in a rectangular spiral as indicated in the Doodling video . For instance, the picture below is a result of one implementation of this idea, with the yellow boxes corresponding to the primes and the blue boxes are the composites – though the first few pixels appear to be discolored:



Start by working out a spiral with pencil and paper and pay attention to what direction you're turning to as the boxes spiral, and how many boxes in a row occur before the next turn and what the pattern of these is. Then try putting together a flow chart indicating values such as "hor," "ver," "inArow," "turns,"

"upordown," and "rightorleft," say. Innovate your own system for following the rectangular spiral.

Your assignment is to write a DarkGDK program that produces an array of bools where the value is 1 if the index is prime and 0 if it's composite (use the sieve of Eratosthenes) and then use these to create an image which shows a rectangular spiraling of these 0/1 values as either one color or another (blue/yellow) in the picture above.

Afterthought #0: Always leave room for afterthought #0.

Afterthought #1: If you're using a width by height window such as the default width = 640, height = 480, then the spiral will start in the center at (320,240). One approach would create an array of `bool box[width][height]` and start by setting

```
box[320][240] = primes[0],(start)
box[321][240] = primes[1],(right)
box[321][239] = primes[2],(up)
box[320][239] = primes[3],(left)
box[319][239] = primes[4],(left)
box[319][240] = primes[5],(down)
box[319][241] = primes[6],(down)
box[320][241] = primes[7],(right)…
```

```
Continue until you suss out the pattern in the directions and
how to index these in a two-dimensional array.  How dimensions
will the dbBox() elements be?  What loop structure will work?
Use additional variables as needed to get the spiral going and
then be sure it stops at a good place.
```

Afterthought #2: To embellishing this a bit, we might also distinguish between composites that are "square free" and those that are not. A natural number is "square free" is it doesn't have any perfect square factors. 18 = 2*32 is not square free, but 30 = 2*3*5 is.

Here's the picture with the square free numbers in magenta. Look at the two black boxes in the middle and count them 0 and 1 going left to right. Then go up (2 is yellow) left (3 yellow) left again (4 blue – not square free) down (5 yellow) down again (6 magenta – square free) right (7 yellow) right (8 blue) right a third time (9 magenta) then up (10 blue) up (11 yellow) up (12 magenta) and so on.