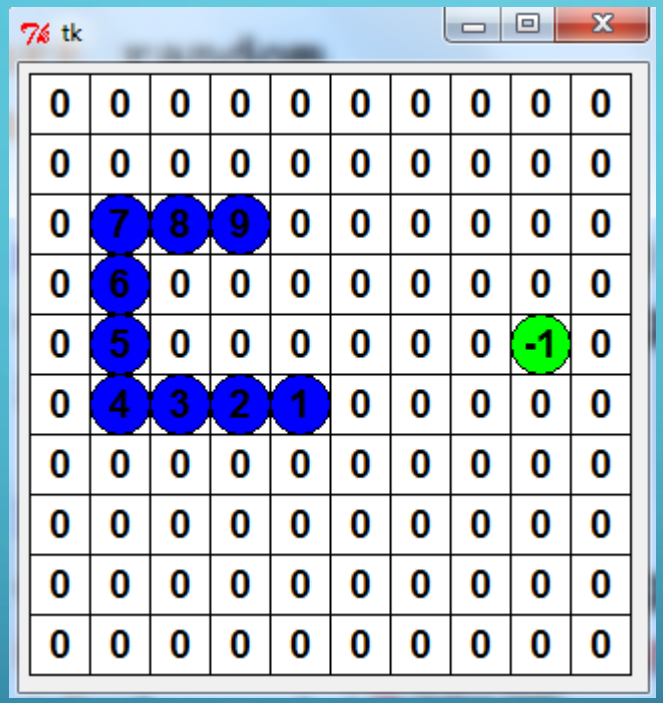
RASPBERRY PI

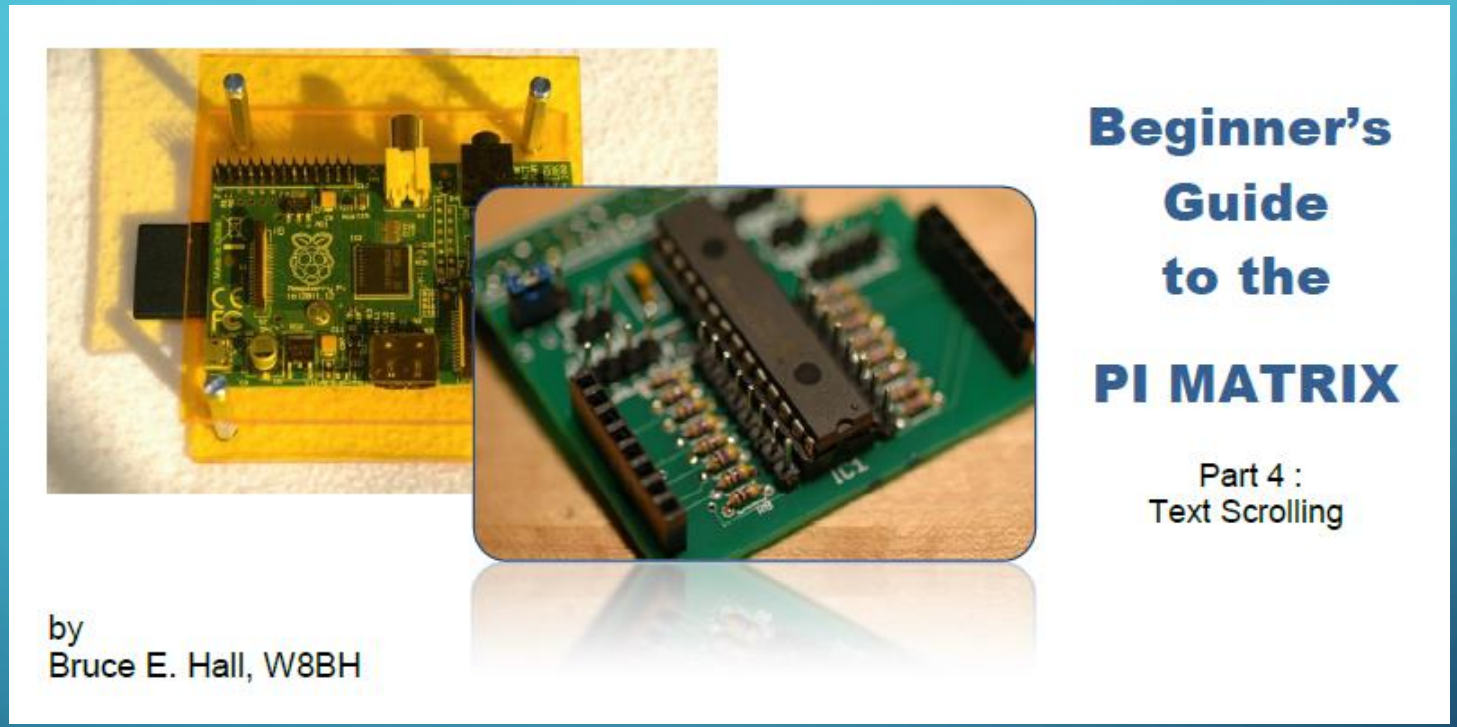
HOW TO UTILIZE A TV MONITOR INSTEAD OF REGULAR LEDS

LEVERAGING THIS SNAKE PROGRAM TO DISPLAY A MATRIX



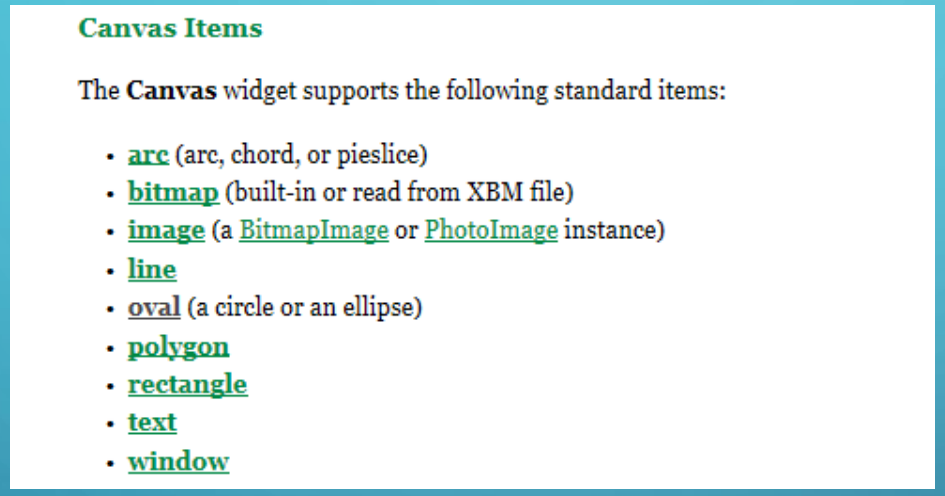
http://www.kosbie.net/cmu/fall-10/15-110/handouts/snake/snake.html

AND COMBINING IT WITH TEXT SCROLLING



HTTP://W8BH.NET/PI/PIMATRIX4.PDF

UTILIZING TKINTER CANVAS TO CREATE THE 'OVAL' PIXELS



Canvas Items

The **Canvas** widget supports the following standard items:

- **arc** (arc, chord, or pieslice)
- **bitmap** (built-in or read from XBM file)
- **image** (a BitmapImage or PhotoImage instance)
- **line**
- **oval** (a circle or an ellipse)
- **polygon**
- **rectangle**
- **text**
- **window**

http://effbot.org/tkinterbook/canvas.htm

PERFORMANCE IS EVERYTHING – CREATE ONCE AND THEN UPDATE

2

You create new items at each updates. The canvas display all the rectangles you have previously added and thus go slower and slower (each update create 900 rectangles, after 30 you have 27,000 objects in your scene...)

To avoid this, you may create your rectangles once, and then only update their colors.

You could have at toplevel:

```
rectangles = [ [ canvas.create_rectangle (CELL_SIZE*x, CELL_SIZE*y,
                    CELL_SIZE*x+CELL_SIZE, CELL_SIZE*y+CELL_SIZE,
                    fill="#000000",outline="#000000", width=1)
                for x in range(nCols)] for y in range(nRows)]
```

and in drawbox:

```
canvas.itemconfig(rectangles[y][x], fill=color)
```

http://stackoverflow.com/questions/10515720/tkinter-canvas-updating-speed-reduces-during-the-course-of-a-program