Programming Beginnings - From Helping the Sighted to Helping the Blind

The start:

Beginners computer savvy but not computer programming savvy

Examples useful but often abstract - Make pictures

**show\_square\_loop\_colors.py** # standard

Display a square with colored sides

Execute program before detailed explanation

A screenshot of a computer

AI-generated content may be incorrect.

Comments

Just a description – “Etch-a-Sketch”

Language of our own – e.g., Best Friends for Ever - BFF, Lots of Luck – LOL

List of colors

For each color in list of colors

Line width(thickness)

Line color

Move 200 pixels

Turn right 90 degrees

**show\_square\_loop\_colors\_w80.py**

Just a small change – thicker lines

Execute program before detailed explanation

A screenshot of a computer

AI-generated content may be incorrect.

**show\_square\_loop\_colors\_braille.py** # braille

The smallest of changes to add braille

A screenshot of a computer

AI-generated content may be incorrect.

We get original drawing

Window simulation of Braille map

Start Notepad.exe to mimic application which sends text to Braille embossing.

Paste to notepad – coarse text map of square figure

First letter for colors r – red, o – orange …

Spaces as commas. Why? Hint: First time I used spaces

**show\_square\_loop\_colors\_w80\_braille.py**

The smallest change– thicker line**.**

**A screenshot of a computer

AI-generated content may be incorrect.**

Note the thicker lines – approximate because of the coarse rendition

Paste the new map into Notepad - can send multiple pictures at once – saves walking

Navigate on Braille window, using numeric pad

4(left), 8(up), 3(right,down), 3, 3

Magnification

Select rectangle enclosing Alt+M,S

A screenshot of a game

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Magnification can be repeated

A screenshot of a computer

AI-generated content may be incorrect.

What can you do?

Our tool is a software program, written in the Python programming language. To use this tool, the user writes or is provided a simple Turtle program to create a simple graphic such as a square. When run this program generates the standard Turtle graphics output, plus text output which when sent to a standard Braille embosser produces a “touchable” rendition of the square.

Our tool, while functional, is a prototype. We could use lots of help.

Our greatest need is a small user community which can use the tool and provide us with feedback as to its usefulness. We can use information as to the tool’s strengths and shortcomings. We would appreciate suggestions on how we can make our tool more useful. Where is it in need of improvement? What are the best points?

The program needs packaging so that we can better distribute it. The software needs testing.

Some options, such as “Expand to Top” were stubbed out, and need work. Experimental features such as perimeter scanning (the audio presentation of the perimeter of the graphics figure), need improvement.

Research, or development, of available, low cost, tools such as a reusable high speed Braille tablet would be great. Integration with high performance touch screens might be helpful.