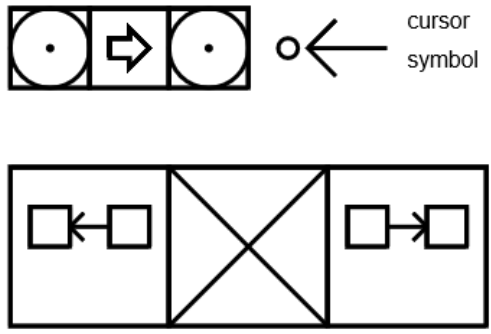

Figure 0.1: cursor edits.

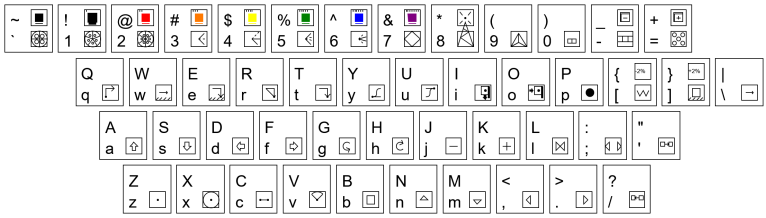

Figure 0.2: keyboard.

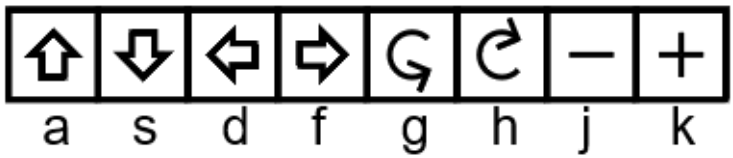

Figure 0.3: Movements. Arrows move along directions of the lines in the cursor. Rotation is by the unit indicated by the cursor wing angles. Scale actions are by the current scale value as shown by the dot positions on the cursor. Letters shown indicate the keys which map to these actions on a QWERTY keyboard with the default settings.

0.1 2d Web Symbols and Icons

Style

Each instance of the Geometron Virtual Machine has a style object, which defines 8 layers, numbered from 0 to 7. Each style has a line color, line width, and fill color. The properties of the style object are stored in the JSON file data/currentjson.txt which is used by the app symbol.html to edit graphics which are used by the rest

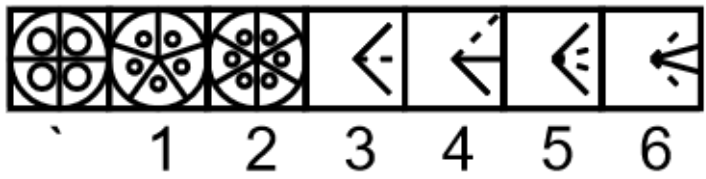

Figure 0.4: Angles described by symmetry glyphs. This also shows the actions to bisect, double, trisect and triple angles, and what keys are used to activate each geometric action.

of the Geometron system.

While the style app edits the data file currentjson.txt which applies to the whole Geometron object used for symbol editing, the importing and exporting of data for sharing with other users only includes style information, without the rest of the JSON data. This allows styles to be separated from the rest of the information for the purposes as usual of building a robust remix culture where Geometron users can constantly be sharing each piece of the system. The EXPORT button will always post the current style JSON in the window in the lower left of the screen. IMPORT will import the data, and RESET returns it to a default state. Try creating your own new

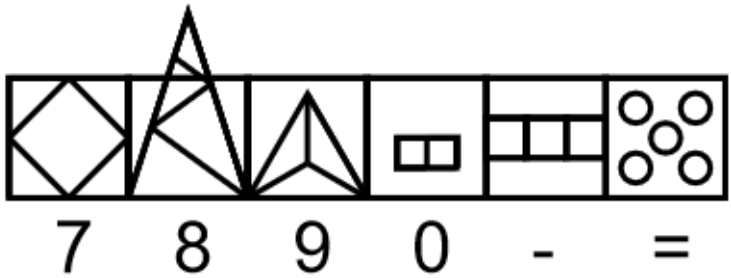

Figure 0.5: Scales, along with keys used to map to them in default configuration. There is no relation between the numbers on the keys and the mathematics of the scales. The scales shown are, from left to right, the square root of 2, the Golden Ratio, the square root of 3, 2, 3, and 5.

style with unusual line widths and colors, then exporting it and saving it offline, sharing it with other users, etc.

Colors are in the format of HTML/CSS/JavaScript, and can be either names of colors like “red” or RGB color values like “#00ff00”. This last format is a number in base 16 which has three 2 digit numbers in it(numbers between 0 and 0xFF), where the three numbers are values of red, green, and then blue. So black is #000000 and white is #FFFFFF. Any value where all three numbers are the same, like #808080 will be a shade of grey. Colors can be partially transparent by adding a fourth hexidec-

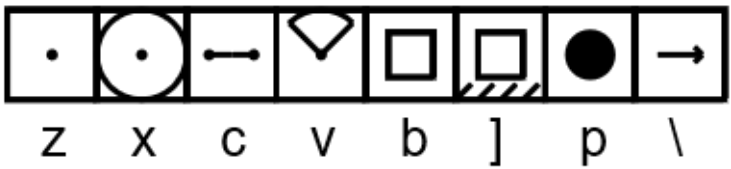

Figure 0.6: Basic drawing actions, along with keys used in default configuration to activate them. From left to right the actions are: draw dot, draw circle of unit radius, draw line segment of unit length, draw arc between cursor wings, draw a square, draw a filled square, draw a filled circle, and draw a line segment while moving forward one unit.

imal number which represents opacity. So fully opaque red is #FF0000FF, and red with half transparency is FF000080(80 because 8 is half of 16, this is actually 128 in decimal).

Graphics Setup

The next section of the JSON we want to know how to edit in order to be able to make useful graphics is the setup, edited in the app setup.html. Setup edits five

Figure 0.7: Layers. Each layer has a line color, line width, and fill color, all of which are set with the Style object using the Style editor app.

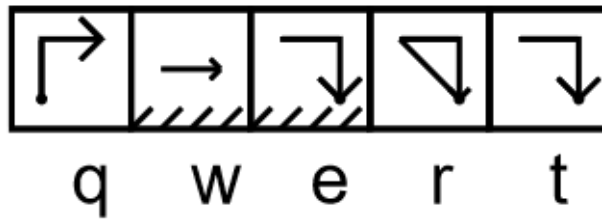

Figure 0.8: Path actions, with keys used to activate them in default state. From left to right, actions are: start path, draw line segment in path, close a filled path, close an unfilled path, and terminate a path without closing it.

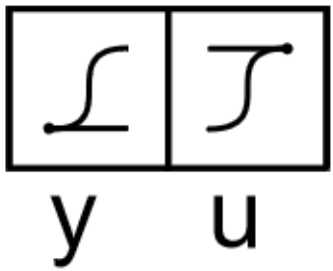

Figure 0.9: Start a Bezier Path and terminate it with the y and u keys.

numbers, all of which are in units of pixels: x0, y0, unit, width and height. Width and height are the width and height of the graphics file currently being edited or created. When a Geometron glyph is drawn with a given GVM, it starts with x and y equal to x0 and y0. Setting these two values is therefore effectively setting the horizontal and vertical offset of the field of view of the symbol. When we activate a pan function within the symbol.html app what we are really doing is modifying the values of x0 and y0 in the JSON file. These are done manually in this app. Finally, unit describes the initial unit value of the GVM. This is essentially the scale factor. So again when we activate the zoom functions in any other symbol editor what we are really doing is making

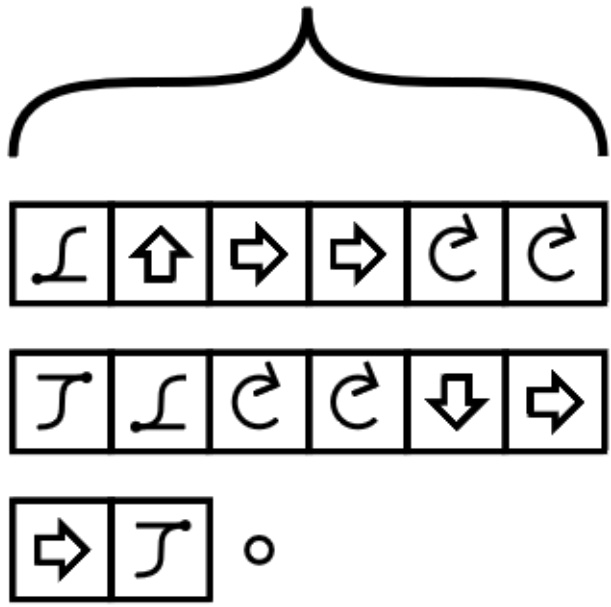

Figure 0.10: Demonstrating the power of Geometron to make useful symbols with Bezier paths quickly and easily: a twiddle bracket.

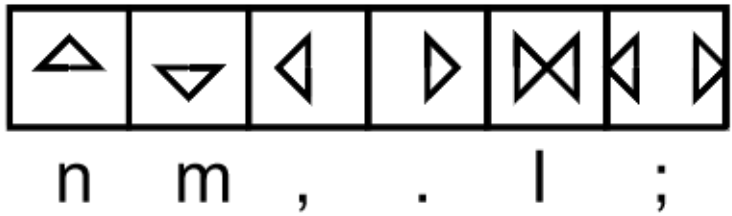

Figure 0.11: Pan and zoom the field of view.

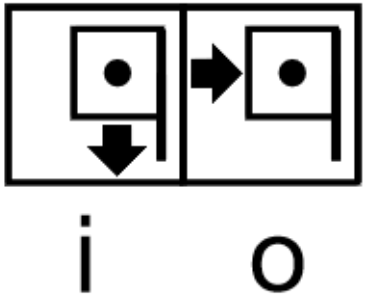

Figure 0.12: Drop a flag, return to flag.

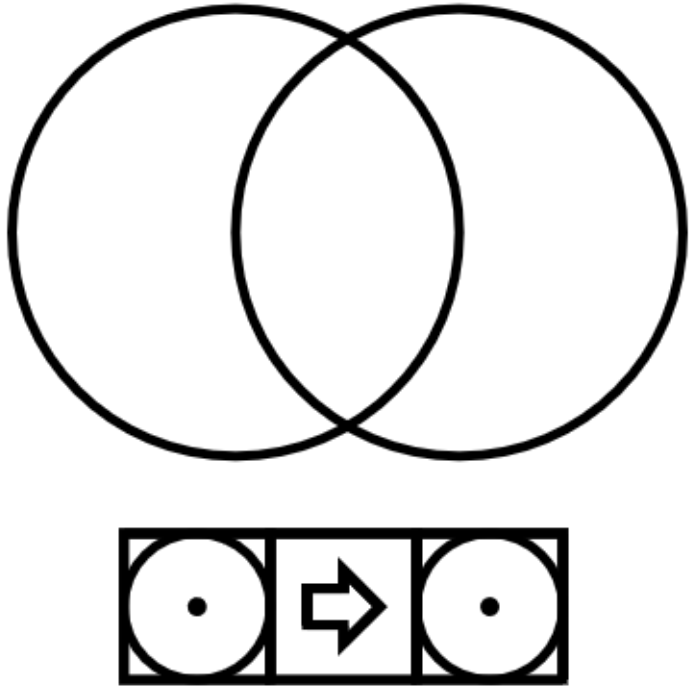

Figure 0.13: The “hello world” of geometric programming, the Vesica Piscis.

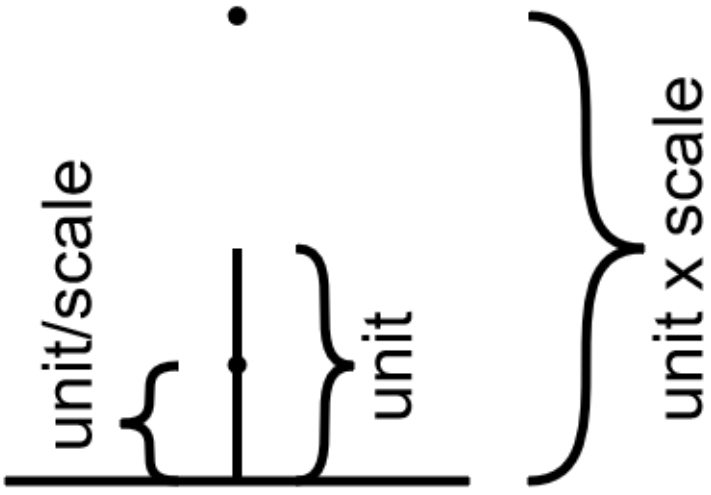

Figure 0.14: Cursor scale.

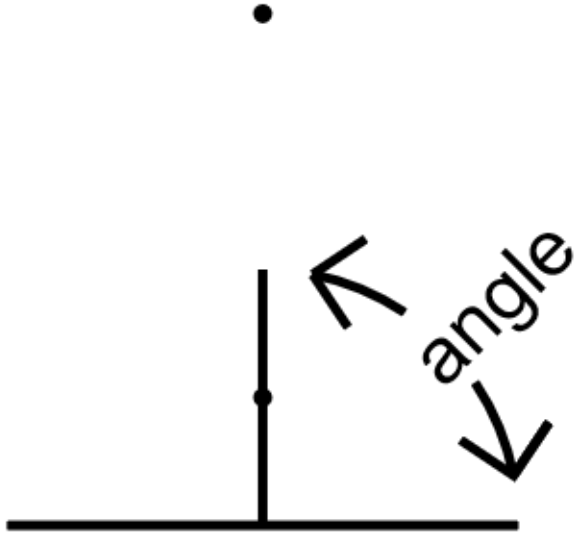

Figure 0.15: Cursor angle.

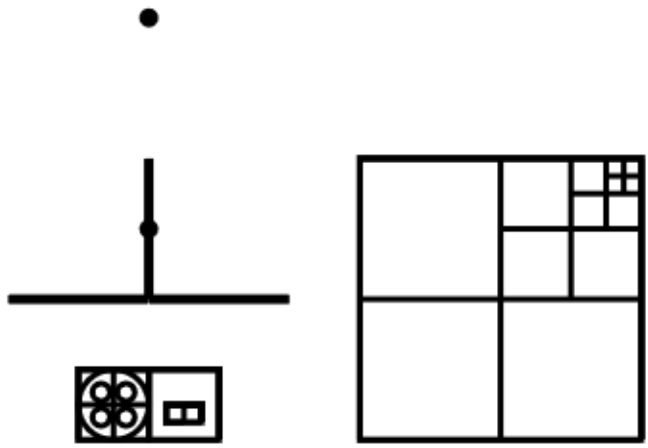

Figure 0.16: Cursor square.

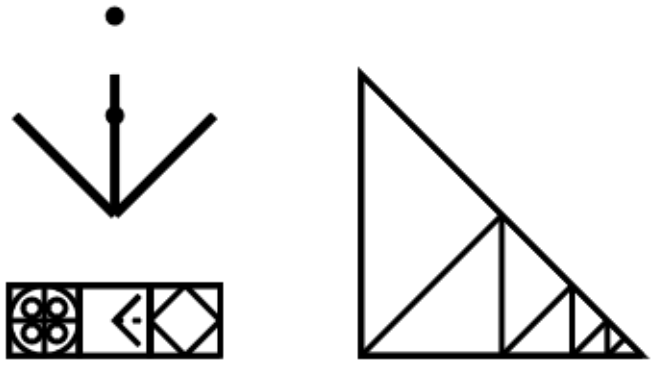

Figure 0.17: Cursor root2.

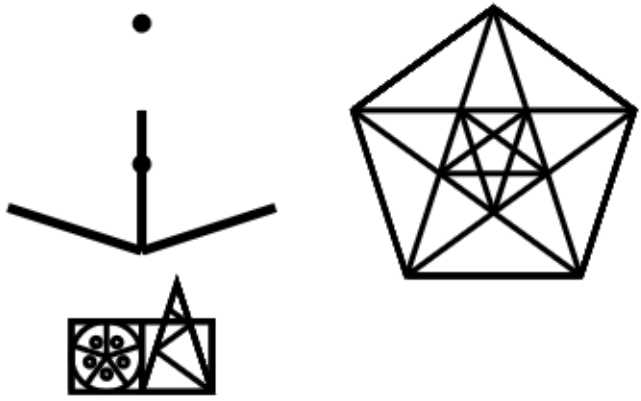

Figure 0.18: Cursor golden ratio.

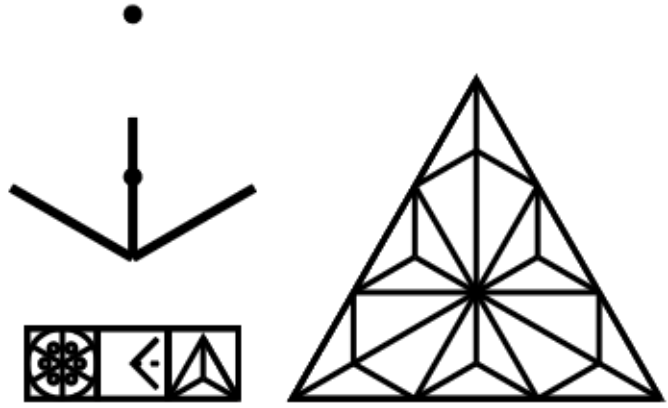

Figure 0.19: Cursor root 3.

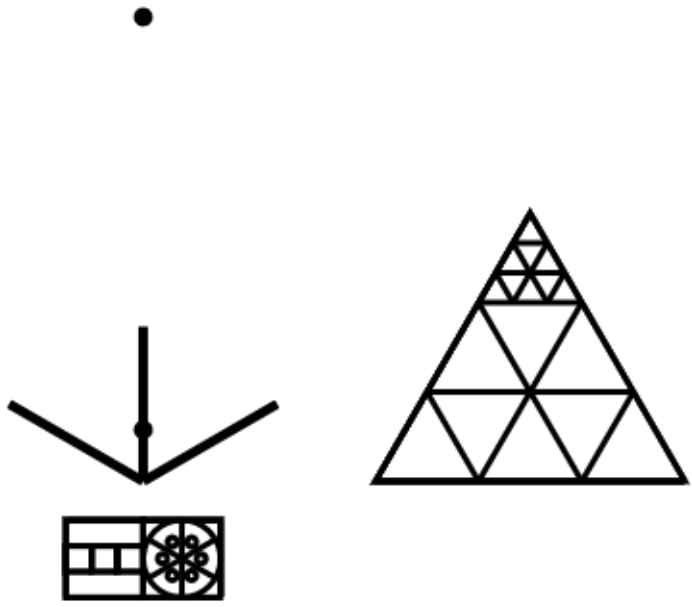

Figure 0.20: Cursor 3.

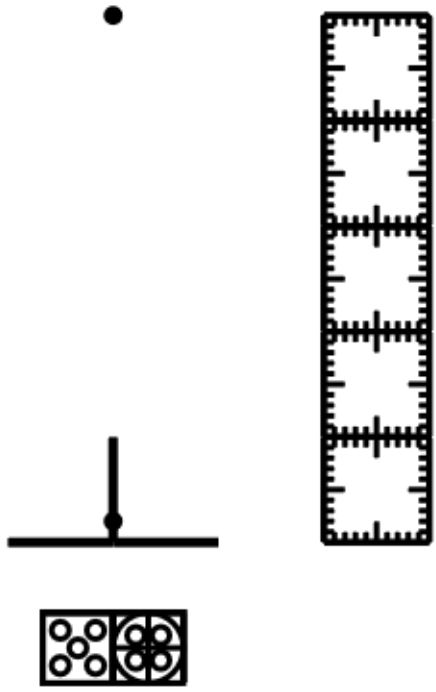

Figure 0.21: Cursor 5.

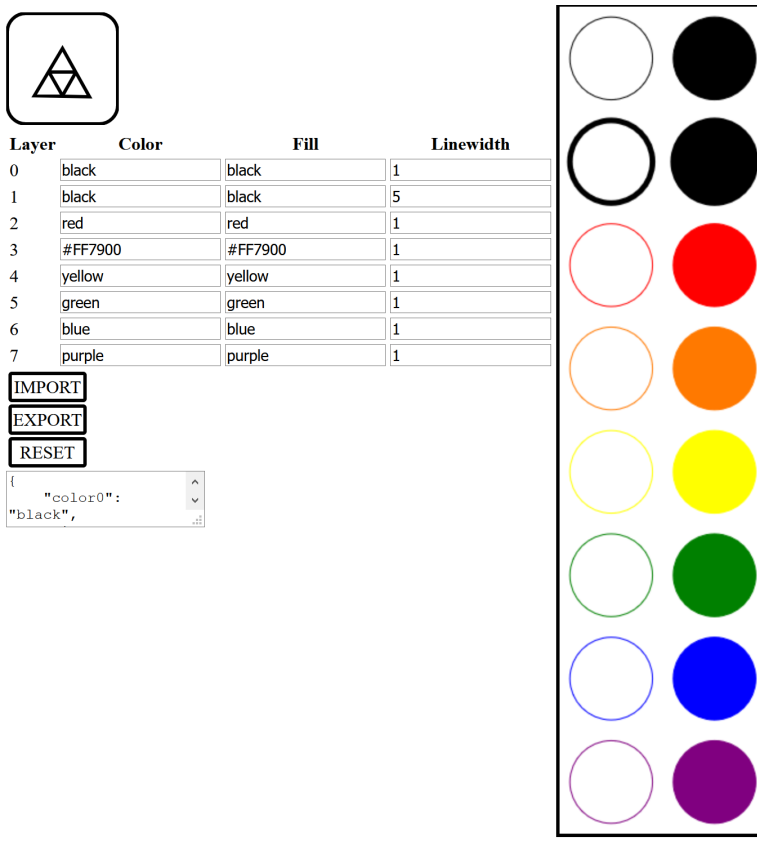

Figure 0.22: Screen shot of the style editor app at styleeditor.html. The display on the right hand side of the screen shows an unfilled circle and filled circle of each layer's style. The text area in the bottom left of the screen is used to import and export style data, which can be saved offline and shared with other users via text message, email, etc. The RESET button resets the style to a standard setting, which will erase any changes made to the existing style. Enter new values into any field to immediately change it.

changes to the variable unit in the global JSON file.

The app setup.html has five fields in which to enter the numbers for the five values. There is also a reset button to restore default, with a 600 by 600 pixel square and 80 pixel unit centered in the center.

The Shape Stack

The final section of the global JSON file which defines the settings of the symbol app is the shape stack. This is a subset of the hypercube, and is stored both in the JSON file data/hypercube.txt and also data/currentjson.txt. When vector graphics files are saved, this shape stack is stored inside them so that they can be reloaded with the whole stack.

Exchange of shape stacks. Manual retrieval from .svg files. uploaded svg files and click to load the whole json.

control panel

hypercube editor for manual direct control

font editor

keyboard editor

laser cut designs

- hello world vesica piscis
- symbols, how they work with hypercube,
- editing, cursor, keyboards, control panels, modes
- symmetries and scales, different methods of geometron(AG)

- cursor,movements, basic constructions(segment, circle, arc, dot)
- layers, colors, lines, style json, working with styles, transparency in hex colors, finding colors
- bezier curves
- paths
- character stack
- fonts
- flags
- tracing symbols from images
- editing the hypercube and shape table, sharing them, import and export of hypercube, sharing of bytecode
- canvas,svg/png/base64 workflow, laser cut shapes production, practical graphics for manuscripts and web, iconsymbols, usage in jupyter notebooks, how the JSON embeds in the SVG, how the symbol feed works, how the setup of JSON works,
- control panels, softkey interfaces, writing geometron apps, how to replicate in other systems from scratch
- examples of using the GVM in JS, documentation of geometron.js, how to use just as a js library to build whatever you want, a list of things someone needs to build to make all this a lot better, how to do that