History Project Write-Up

James Gillespie

November 17, 2022

The idea behind my project is to demonstrate the compounding nature of Euclid's constructions. This is achieved through a web app I made called (I need a name). The nature of how I've made this is that, in programming a new construction, I am forced to use only Plato's Axioms and previously completed constructions. In fact, besides the Axioms, I am not allowed to program at all - instead, I constrain myself to writing a JSON formatted list of commands - each element is either a direct use of an Axiom, or a previous construction - to complete the construction.

The framework I made to allow for this style of "programming" uses JSON objects to store the commands for a construction. For example, the construction for a circle might look something look this (the exact format may have changed by now):

```
 1   "name": "circle",
 2           "fargs": [0,0,0,0],
 3           "list": [
 4               {
 5                   "jname": "point",
 6                   "args": [0, 0],
 7                   "farg": [0, 1],
 8                   "name": "P"
 9               },
10               {
11                   "jname": "point",
12                   "args": [1,1],
13                   "farg": [2, 3],
14                   "name": "Q"
15               },
16               {
17                   "jname" : "circle",
18                   "args" : ["P", "Q"],
19                   "farg" :[],
20                   "name": "A"
21               }
22           ],
```

The element "list" stores the commands needed for the construction. list[0] declares a point P located at either $(0,0)$ or defined by arguments passed in elsewhere, using the "Axiom function" point(). list[1] similarly defines a point Q, and the "Axiom function" circle() is invoked to draw a circle centered at P, passing through Q.

Objects such as P and Q are stored in a list where they can be looked up by name - commands such as the call to "circle" require access to them. The circle itself will also be present in this list, allowing more complicated constructions to use "circle" as a command too.