

```

1 // src/main.cc
2
3 #include <iostream>
4
5 #include "hyperfractal.hh"
6 #include "guimain.hh"
7 #include "utils.hh"
8
9 using namespace std;
10
11 /**
12  * Naming Convention:
13  * Classes & Structs - CapitalisedCamelCase
14  * Variables - snake_case
15  * Functions - uncapitalisedCamelCase
16  * Constants - SCREAMING_SNAKE_CASE
17  *
18  */
19
20 int main (int argc, char *argv[]) {
21     if (argc == 8) {
22         // If we have the required arguments, run a console-only render
23         HFractalMain hm;
24         int argument_error = 0;
25         try {
26             hm.setResolution (stoi (argv[1]));
27             if (hm.getResolution() <= 0) throw runtime_error("Specified resolution
too low.");
28             argument_error++;
29             hm.setOffsetX (stod (argv[2]));
30             argument_error++;
31             hm.setOffsetY (stod (argv[3]));
32             argument_error++;
33             hm.setZoom (stod (argv[4]));
34             argument_error++;
35             hm.setEquation (string (argv[5]));
36             if (!hm.isValidEquation()) throw runtime_error("Specified equation is
invalid.");
37             argument_error++;
38             hm.setWorkerThreads (stoi (argv[6]));
39             if (hm.getWorkerThreads() <= 0) throw runtime_error("Must use at least
one worker thread.");
40             argument_error++;
41             hm.setEvalLimit (stoi (argv[7]));
42             if (hm.getEvalLimit() <= 0) throw runtime_error("Must use at least one
evaluation iteration.");
43             argument_error++;
44             hm.generateImage(true);
45             return !hm.autoWriteImage (IMAGE_TYPE::PGM);
46         } catch (runtime_error e) {
47             cout << "Parameter error on argument number " << argument_error << ":" <<
endl;
48             cout << " " << e.what() << endl;
49             return 1;
50         }
51     } else if (argc != 1) {
52         // If we have only some arguments, show the user what arguments they need to
provide
53         cout << "Provide all the correct arguments please:" << endl;

```

```
54         cout << "int resolution, long double offset_x, long double offset_y, long
double zoom, string HFractalEquation, int worker_threads, int eval_limit" << endl;
55         return 1;
56     } else {
57         // Otherwise, start the GUI
58         return guiMain(argv[0]);
59     }
60 }
```