

# How to Draw Fractals with C

Matthew James Lindsay

June 21, 2022

# Contents

<b>1</b>	<b>Elementary Math</b>	<b>2</b>
1.1	Sets . . . . .	2
1.2	Some useful sets . . . . .	2
1.3	Functions . . . . .	2
1.4	Induction . . . . .	2
<b>2</b>	<b>C and the Shell</b>	<b>2</b>
2.1	Hello World . . . . .	2
2.2	defining functions . . . . .	3
2.3	predicate logic . . . . .	3
<b>3</b>	<b>Drawing Geometry</b>	<b>3</b>
3.1	Cartesian Geometry . . . . .	3
3.2	Drawing dots . . . . .	4
3.3	for loops to draw lines . . . . .	4
<b>4</b>	<b>Fractal Geometry</b>	<b>4</b>
4.1	Cantors Comb . . . . .	4
4.2	Circles in circles . . . . .	4
4.3	Mandelbrot set . . . . .	4
	<b>Appendices</b>	<b>4</b>
<b>A</b>	<b>Useful code</b>	<b>4</b>

This set of notes assume no maths higher than GCSE level and absolutely no programming experience.

## 1 Elementary Math

### 1.1 Sets

sets can be defined by listing its elements separated by commas in curly braces ie  $S = \{1, 2, 3, 4\}$

Two sets A and B are equal iff every element of A is in B, and every element of B is in A. ie if  $A = \{1, 2, 3, 4\}$  and  $B = \{1, 2, 3, 4, 4\}$  then  $A = B$  (repeats aren't counted in a set)

### 1.2 Some useful sets

$\mathbb{N}$  the set of natural numbers  $\mathbb{N} = \{0, 1, 2, 3, \dots\}$

$\mathbb{Z}$  the set of all integers  $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, 3, \dots\}$

$\mathbb{Q}$  the set of rational numbers  $\mathbb{Q} = \{\frac{a}{b} : a, b \in \mathbb{Z}, b \neq 0\}$

$\mathbb{R}$  the set of real numbers

$\mathbb{C}$  the set of complex numbers  $\mathbb{C} = \{a + bi : a, b \in \mathbb{R}\}$

### 1.3 Functions

functions map every element from one set of elements(domain) to an element of another set(codomain). for example the function  $f(x) = x^2$  maps the number 2 to 4 and -3 to 9. here the domain is  $\mathbb{R}$  and the codomain is  $\mathbb{R}^+$

### 1.4 Induction

## 2 C and the Shell

In this course we will be using the C programming language because C is the bestest!

### 2.1 Hello World

Type the code in the following box into your favorite text editor

```
#include <stdio.h>
```

```
int main() {  
    printf("Hello, World!\n");  
}
```

save as hello.c

at the command line type

```
gcc hello.c && ./a.out
```

this is really two commands, the first `gcc hello.c` compiles our source code into machine language. this outputs the file `a.out` in the same directory as the source code.

## 2.2 defining functions

In C functions are defined by writing the return type, the function name, and have the variables separated by commas in parentheses.

MAJOR DIFFERENCE between C functions and maths functions, C functions have SIDE EFFECTS!!!.

## 2.3 predicate logic

A	B	$A \cap B$	$A \cup B$
F	F	F	F
F	T	F	T
T	F	F	T
T	T	T	T

# 3 Drawing Geometry

## 3.1 Cartesian Geometry

2D Cartesian geometry is used to encode and manipulate points in 2D space to do this we choose some point as the origin and mark in O, choose a unit vector in some direction

Now in computer graphics the origin is the top left pixel of the monitor and

## 3.2 Drawing dots

The bare minium code necessary to draw a window and draw a dot to it.

```
#include <SDL2/SDL.h>

int main() {
    SDL_Init(SDL_INIT_VIDEO);
    SDL_Window* window = SDL_CreateWindow(
        "draw dot", SDL_WINDOWPOS_UNDEFINED,
        SDL_WINDOWPOS_UNDEFINED, 1000, 1000, 0);
    SDL_Renderer* renderer = SDL_CreateRenderer(
        window, -1, SDL_RENDERER_ACCELERATED);

    SDL_SetRenderDrawColor(renderer, 0, 0, 0, 0);
    SDL_RenderClear(renderer);

    // clean up
    SDL_DestroyRenderer(renderer);
    SDL_DestroyWindow(window);
}
```

### 3.3 for loops to draw lines

## 4 Fractal Geometry

### 4.1 Cantors Comb

### 4.2 Circles in circles

### 4.3 Mandelbrot set

# Appendices

## A Useful code

```
class Complex { public:
    double a, b;
    Complex(double a_, double b_) :a{a_}, b{b_} {}

    Complex operator + (Complex const& obj) {
        return Complex(a+obj.a, b+obj.b);
    }
};
```