

COMP140 WORKSHEET B: FRACTALS

Version 1.0
BSc Computing for Games
COMP140

Brian McDonald & Ed Powley

In this worksheet, you will use the SDL2 library (<https://www.libsdl.org/index.php>) to write a program to generate and display a fractal; see Figure 1. This fractal colours each pixel of the image according to an iterated mathematical formula, as described below. The following GitHub repository contains a project named `Fractal` for you to build upon.

<https://github.com/Falmouth-Games-Academy/comp140-worksheetB>

This contains code to create and display a random image; you will implement the calculations to generate the fractal.

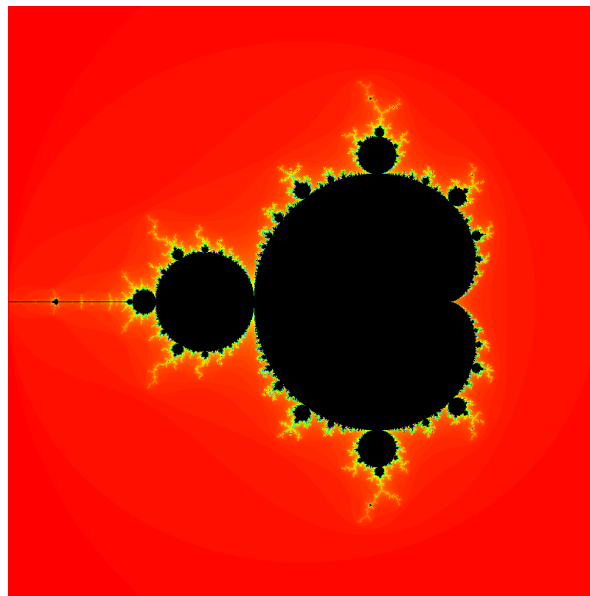


Figure 1: An example of a fractal

1

To generate an interesting fractal, the on-screen x and y coordinates must first be rescaled. In the skeleton project the pixel coordinates range from 0 to 800, whereas the fractal is most interesting in the region $-2 \leq x \leq 1$ and $-1.5 \leq y \leq 1.5$.

Let p_x be the x coordinate of the pixel. This can be remapped into the range x_{\min} to x_{\max} using the following formula:

$$x_0 = \frac{p_x}{\text{image.width}} \times (x_{\max} - x_{\min}) + x_{\min}$$

The y coordinate can be remapped using a similar formula.

Implement the above calculations for the x and y coordinates, at the indicated parts of `Fractal.cpp`.

2

The fractal set is based on the following sequence of numbers. Let x_0 and y_0 be the coordinates of a point in the image. Then the sequence $x_1, y_1, x_2, y_2, x_3, y_3, \dots$ is defined¹ recursively for $i = 0, 1, 2, 3, \dots$ by:

$$\begin{aligned}x_{i+1} &= (x_i)^2 - (y_i)^2 + x_0 \\y_{i+1} &= (2 \times x_i \times y_i) + y_0\end{aligned}$$

The points are coloured according to the *smallest* value of i for which $(x_i)^2 + (y_i)^2 \geq 4$. If such a value of i is not found after a large number of iterations (for example $i = 200$), the pixel is coloured black.

Implement an algorithm which performs the above computation, determining the smallest value of i for which $(x_i)^2 + (y_i)^2 \geq 4$ and selecting the appropriate pixel colour.

Implement the algorithm in `main.cpp` so that the program generates the fractal (Figure 1) when it is run.

Submission instructions

You should submit this project as part of the final coursework submission.

Marking criteria

Remember that **it is better to submit incomplete work than to submit nothing at all**.

To demonstrate **basic competency**, complete the following:

- **Timely Submission:** Obtain the marks for timely submission, you must submit (as a GitHub pull request). As with other worksheets, you may resubmit after these deadlines in order to collect extra correctness or quality marks. This is awarded as long as you submit *something* for each part by the deadline, even if your submission has bugs or other issues.
- Some evidence of emerging innovation and/or creativity in the design of the controller and game.

To demonstrate **basic proficiency**, complete the following:

- **Achieve basic competency**
- **Complete** Algorithm 1. **Note:** You will not be penalised for trivial errors which do not affect the overall functioning of your programs
- Appropriate use of GitHub, with descriptive commit messages
- Comments are used where appropriate, and are well written.
- Little evidence of emerging innovation and/or creativity in the design of the controller and game.

To demonstrate **novice competency**, complete the following:

¹If you are familiar with complex numbers, you may notice that this is equivalent to $z_{j+1} = z_j^2 + z_0$, where $z_j = x_j + y_j i$.

- Achieve **basic proficiency**
- **Complete** Algorithm 2. **Note:** You will not be penalised for trivial errors which do not affect the overall functioning of your programs
- Your code is well formatted. Variable and function names are clear and descriptive.
- Much evidence of emerging innovation and/or creativity in the design of the controller and game.

To demonstrate **novice proficiency**, complete the following:

- Achieve **novice competency**
- **Research** a different fractal
- Considerable evidence of mastery of innovative and creative practice in the design of the controller and game.

To demonstrate **professional competency**, complete the following:

- Achieve **novice proficiency**
- **Implement** a new fractal and allow the user to choose what fractal to generate
- Significant evidence of mastery of innovative and creative practice in the design of the controller and game.