MANDELBROT FRACTAL PLOTTER AND JULIA SET PLOTTER.

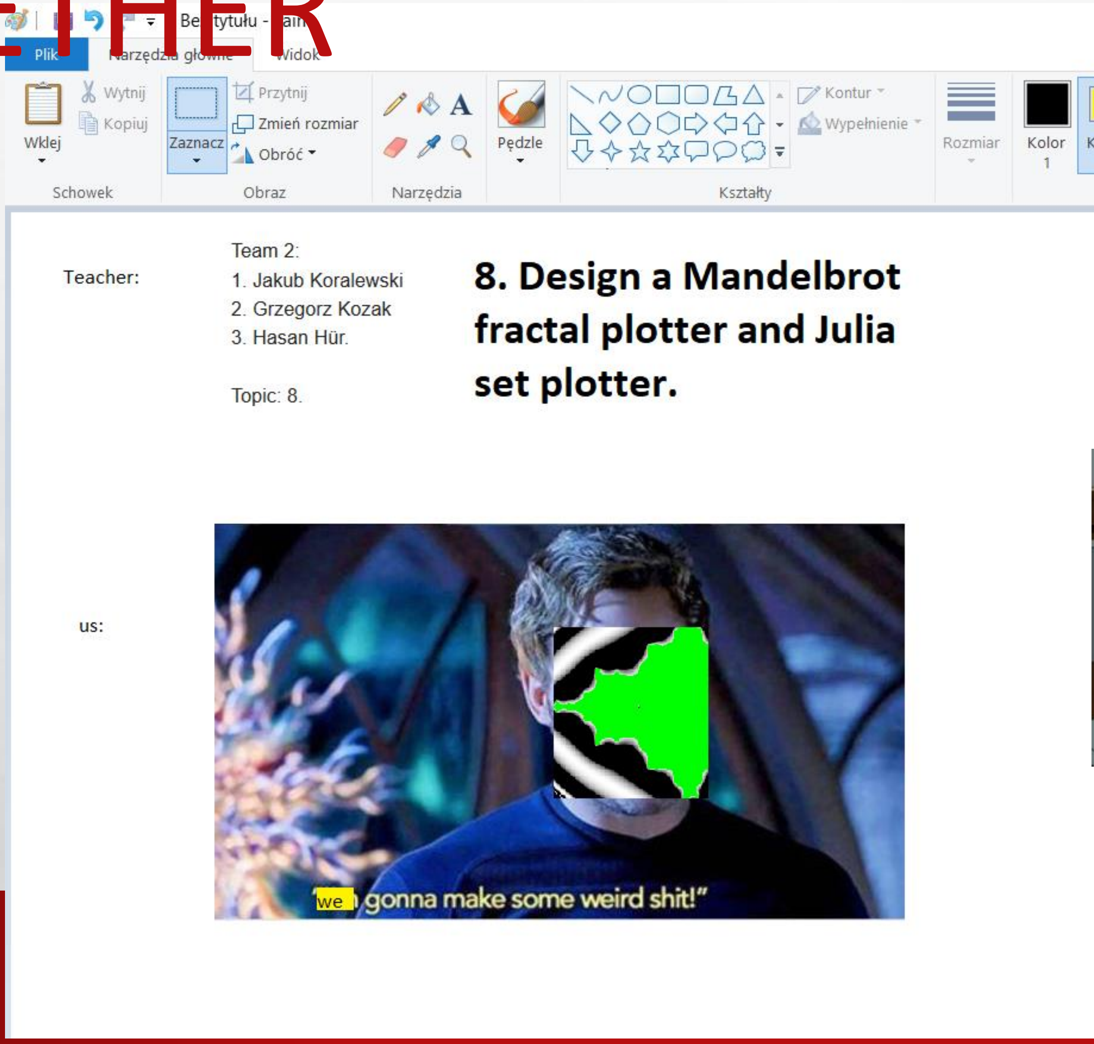
THEORETICAL INTRODUCTION TO TOPIC

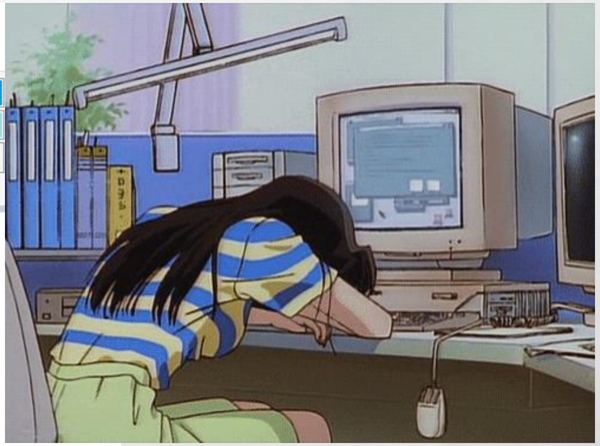
JAKUB KORALEWSKI

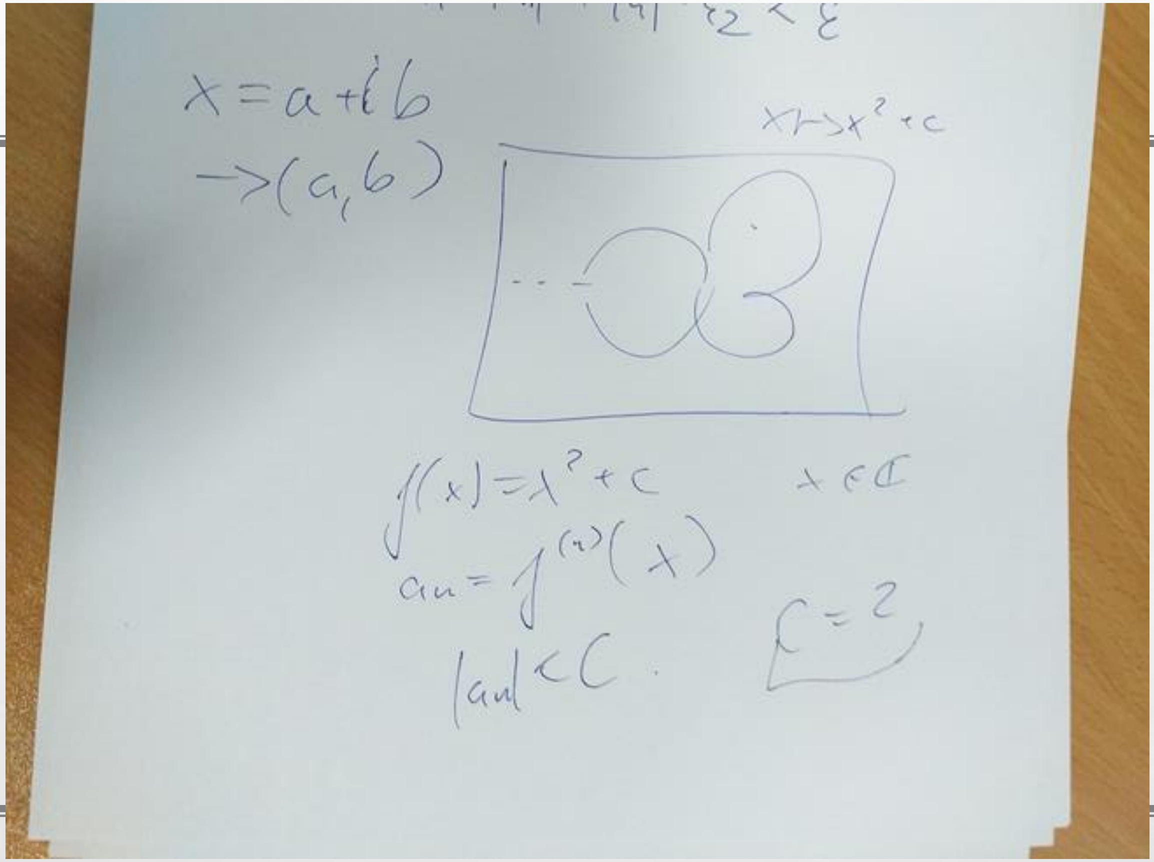
GRZEGORZ KOZAK

HASAN HÜR

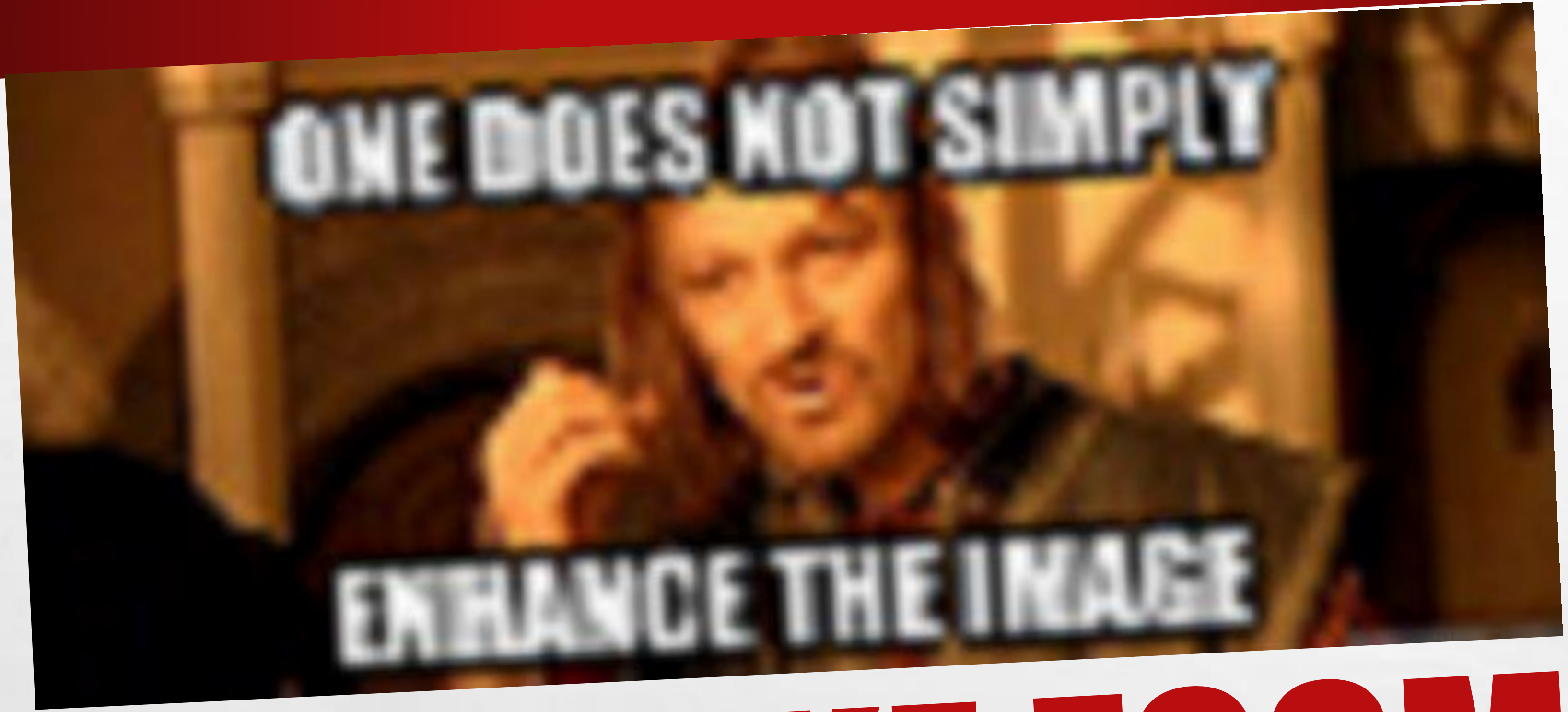
THAT IS HOW WE WORKED TOGETHER



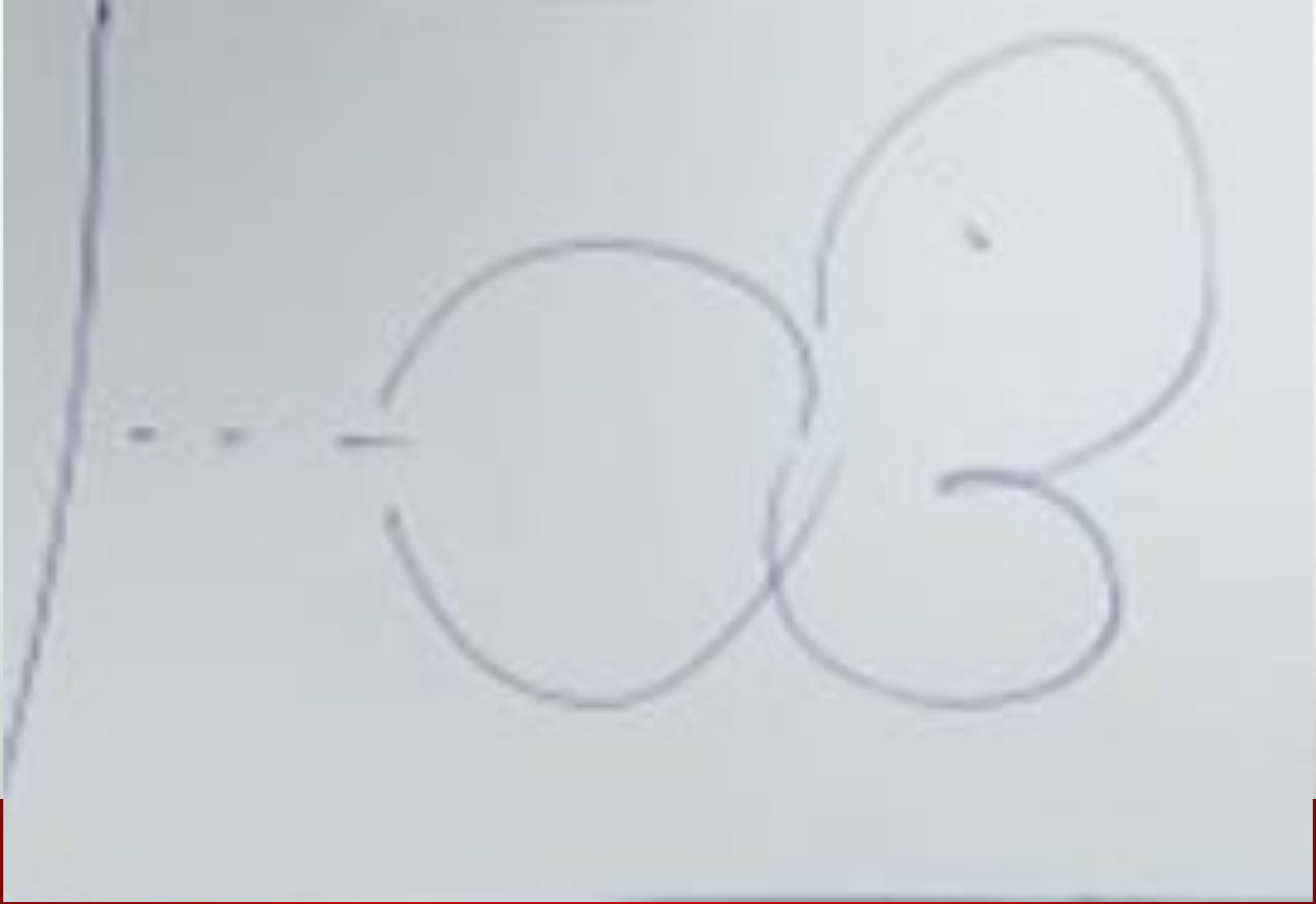


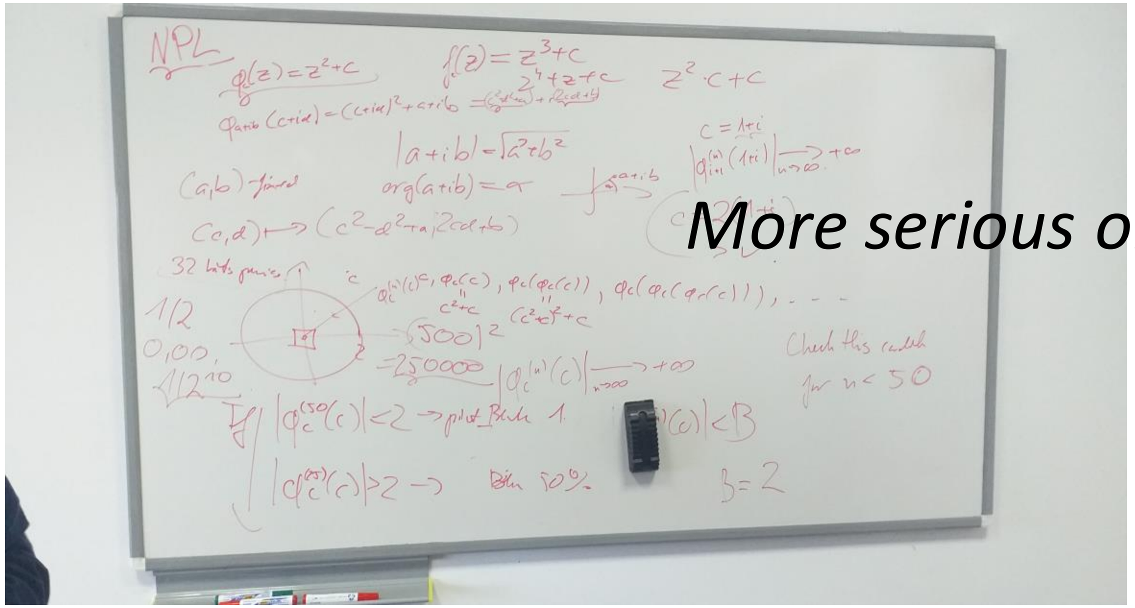


AND THIS IS THE FIRST MEETING WITH DR. NASKRĘCKI CONCERNING OUR PRESENTATION

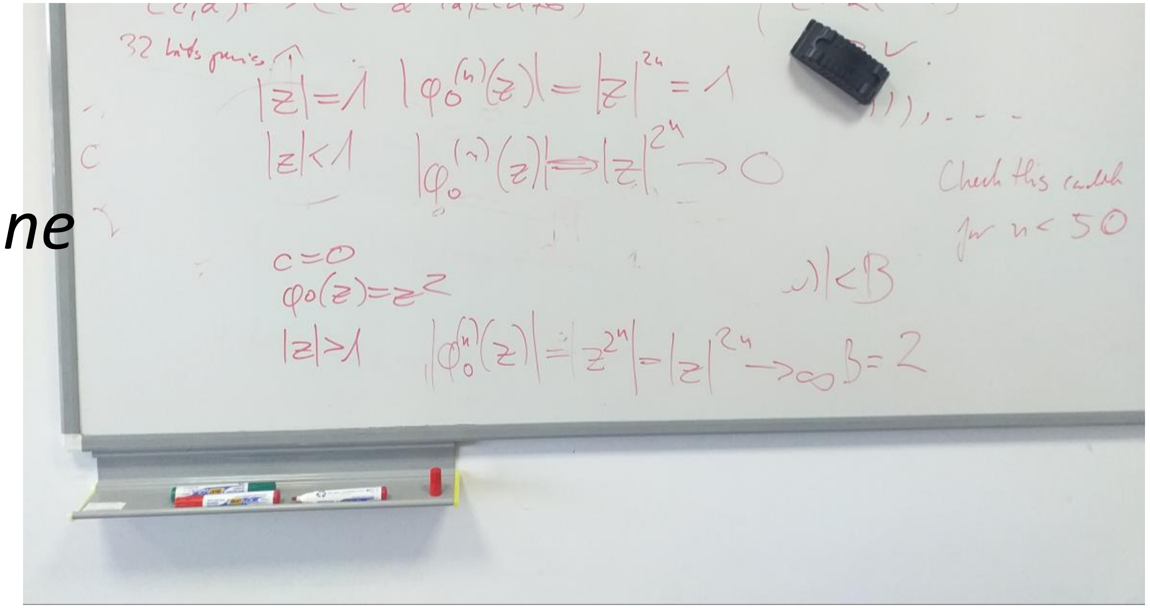


LET'S TAKE ZOOM ON IT





More serious one



THIS IS OUR SECOND CONSULTATION

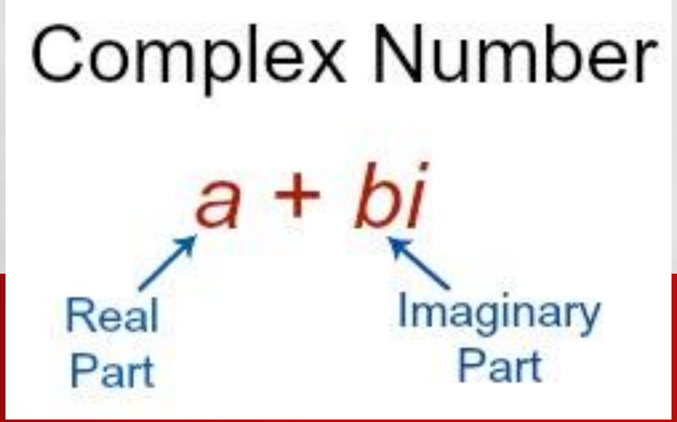
WHAT IS MANDELBROT SET?

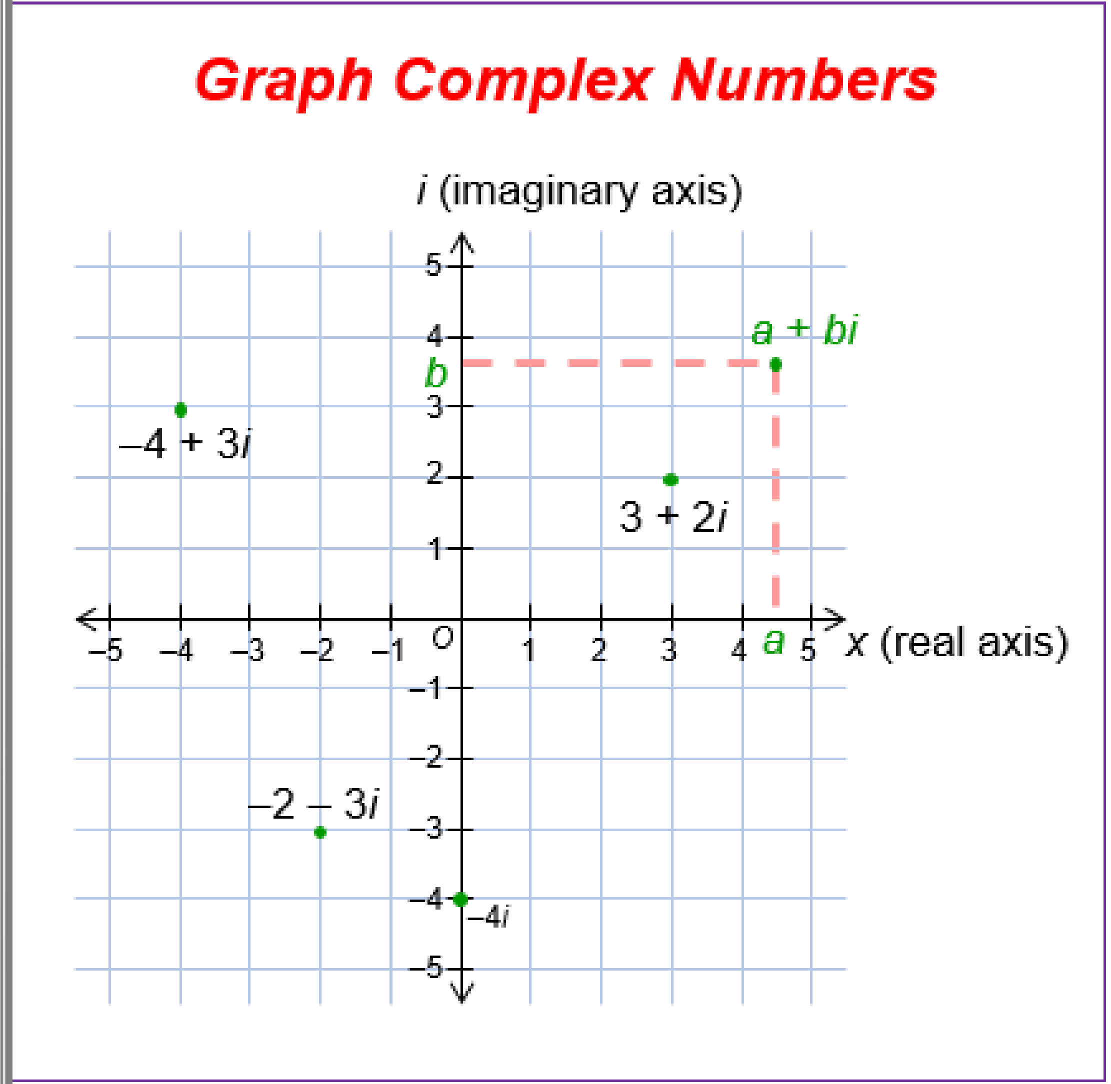
LET'S GET SERIOUS, LADIES

MANDELBROT SET IS THE SET OF COMPLEX NUMBERS

To remind you what complex number is

- SO TO CHARACTERIZE A SINGLE COMPLEX NUMBER, WE NEED TWO AXES, SO OUR CODE , WHICH DRAWS MANDELROT GRAPH WILL BE 2-DIMENSIONAL, AS YOU SEE LATER
- I IS CONSTANT WHERE I^2 = -1, A & B ARE REAL NUBERS





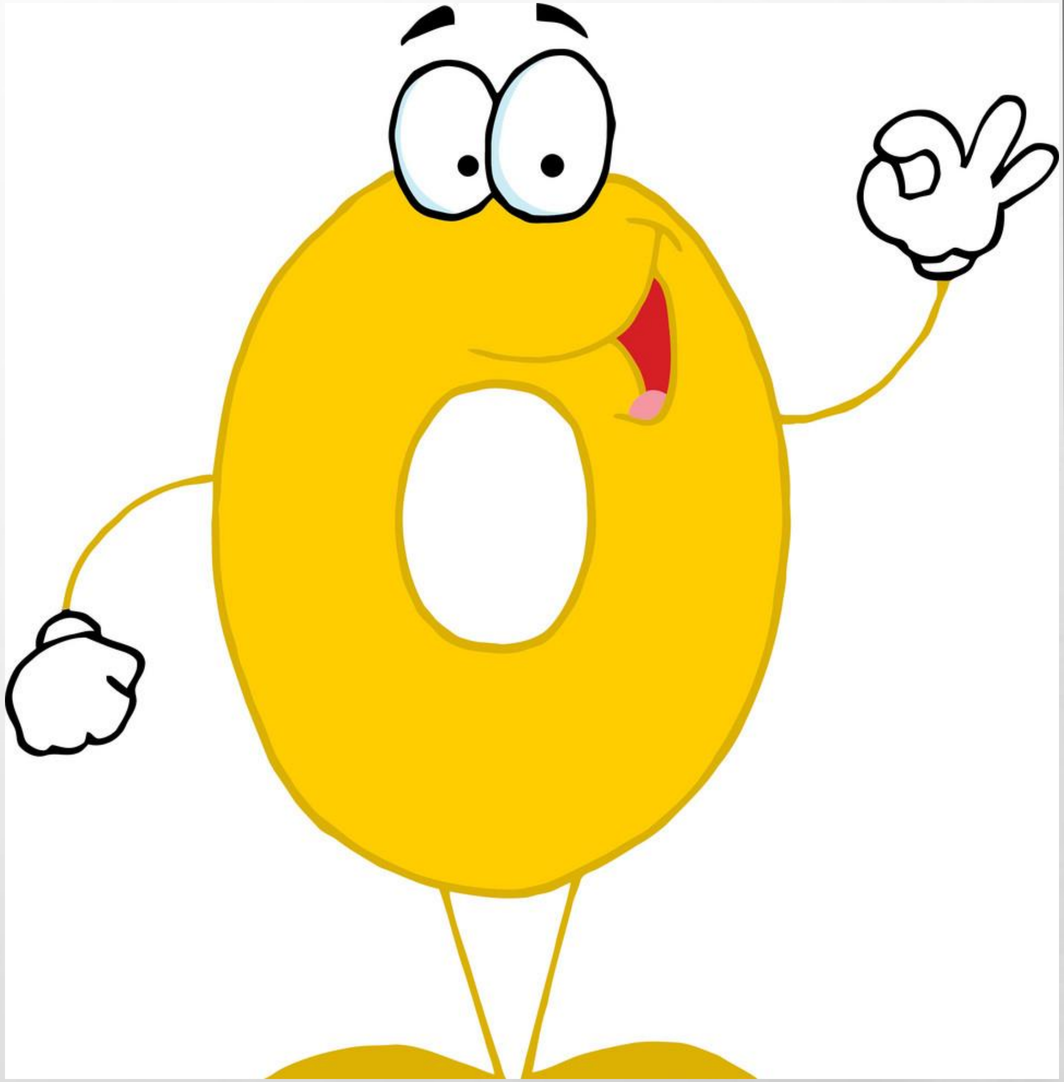
IT IS ALL ABOUT ITERATION AND SQUARING

THE MANDELBROT SET IS THE SET OF COMPLEX NUMBERS C FOR WHICH THE FUNCTION

$$f_c(z) = z^2 + c$$

DOES NOT DIVERGE WHEN ITERATED FROM

$$z = 0$$



LET US TRY FOR C = 1

$$Fx_2(0) = 0^2 + 1 = 1$$

$$Fx_2(1) = 1^2 + 1 = 2$$

$$Fx_2(2) = 2^2 + 1 = 5$$

$$Fx_2(5) = 5^2 + 1 = 26$$

$$Fx_2(26) = 26^2 + 1 = 677$$

$$Fx_2(677) = 677^2 + 1 = 458\ 330$$

$$Fx_2(458\ 330\ ) = 458\ 330\ ^2 + 1 = 210\ 066\ 388\ 901$$

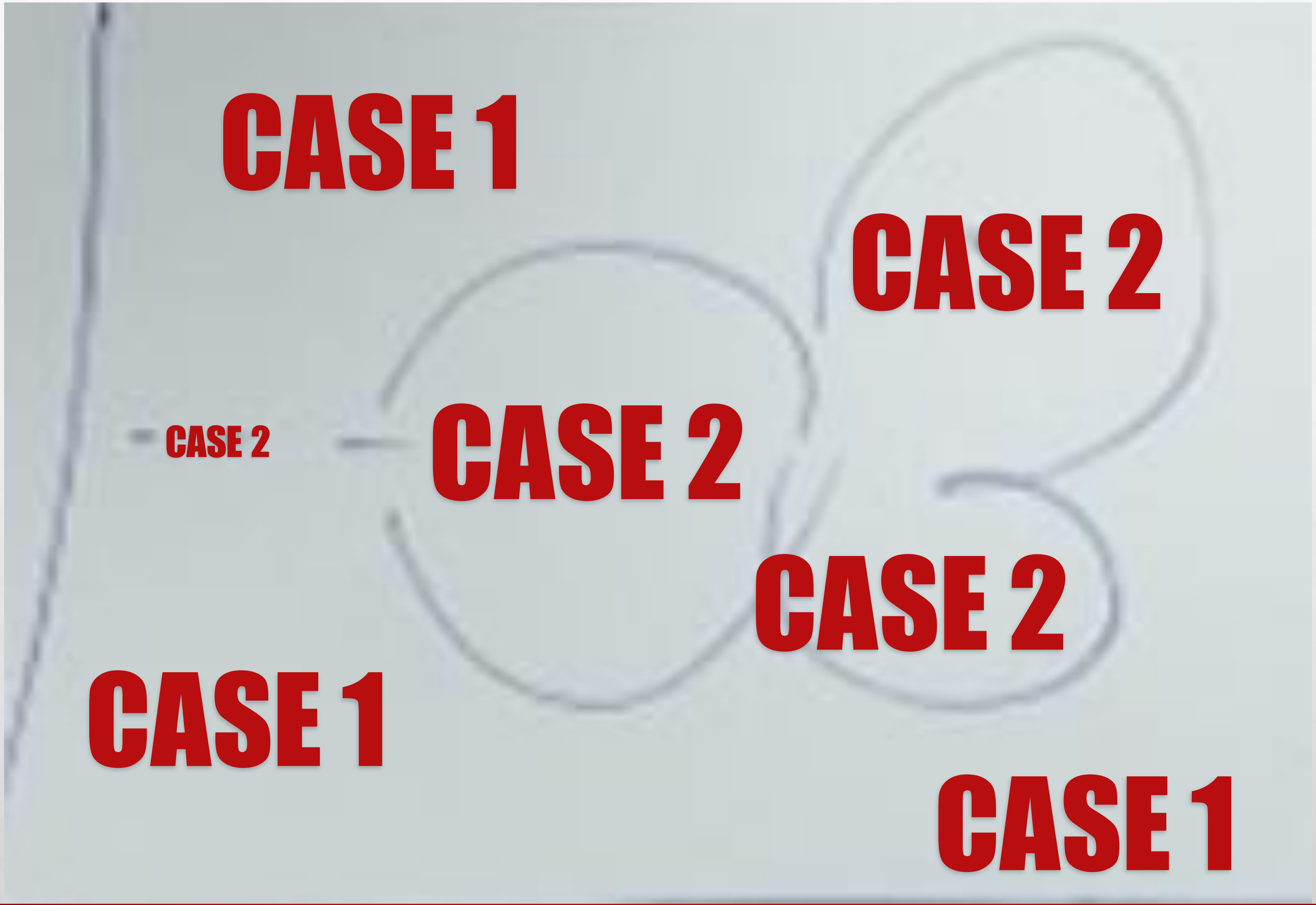
IT GETS LARGER AND LARGER SO FAST

CASE 1

CASE 2: DISTANCE(FROM 0) IS BOUNDED

Let us try for c = -1

$$Fx_{-1}(0) = 0^2 - 1 = -1$$
$$Fx_{-1}(-1) = (-1)^2 - 1 = 0$$
$$Fx_{-1}(0) = 0^2 - 1 = -1$$
$$Fx_{-1}(-1) = (-1)^2 - 1 = 0$$
$$Fx_{-1}(0) = 0^2 - 1 = -1$$
$$Fx_{-1}(-1) = (-1)^2 - 1 = 0$$


CASE 1
CASE 2
CASE 2
CASE 2
CASE 2
CASE 1
CASE 1

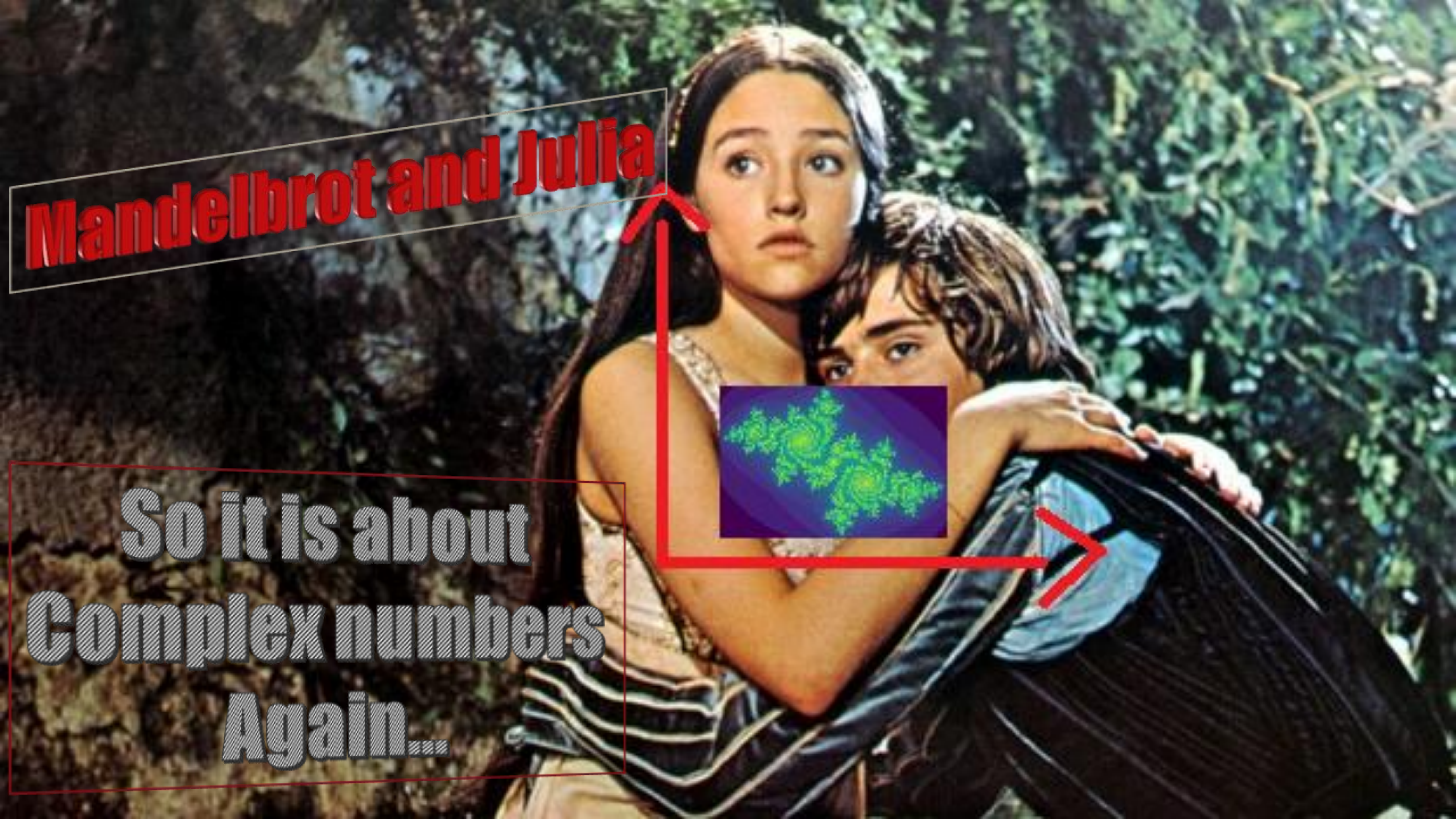
A LITTLE OF HISTORY FOR DUMMIES

THE MANDELBROT SET:

- IS NAMED AFTER BENOÎT MANDELBROT,
  A POLISH-FRENCH-AMERICAN MATHEMATICIAN
- IS IMPORTANT FOR THE CHAOS THEORY
- WAS ONE OF THE FIRST TO USE COMPUTER GRAPHICS TO CREATE AND DISPLAY FRACTAL GEOMETRIC IMAGES
- FIRST DEFINED AND DRAWN IN 1978

JULIA SET

JULIA SET IS NAMED AFTER SIR GASTON JULIA NOT LADY JULIA, LADIES


Mandelbrot and Julia
So it is about
Complex numbers
Again...

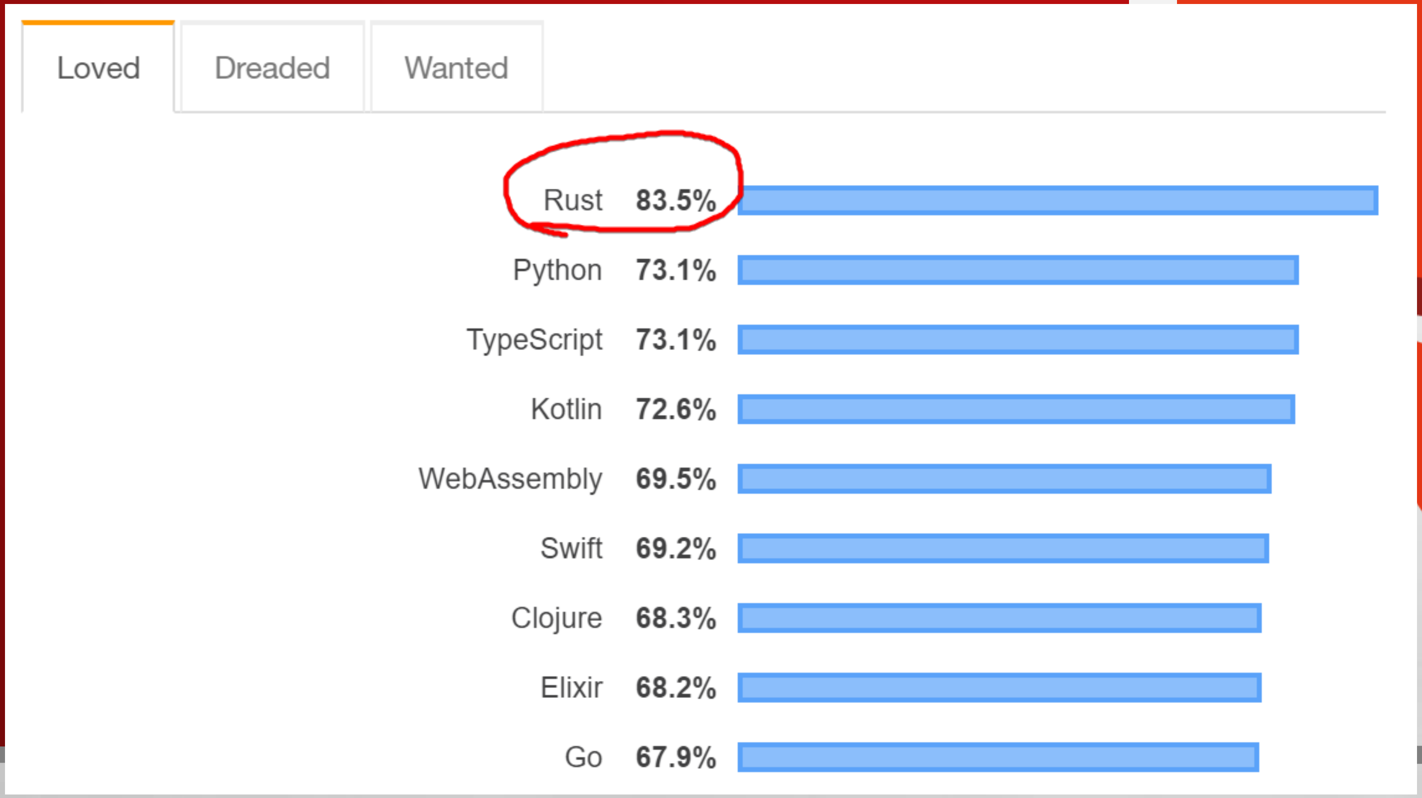
TECHNICAL IMPLEMENTATION

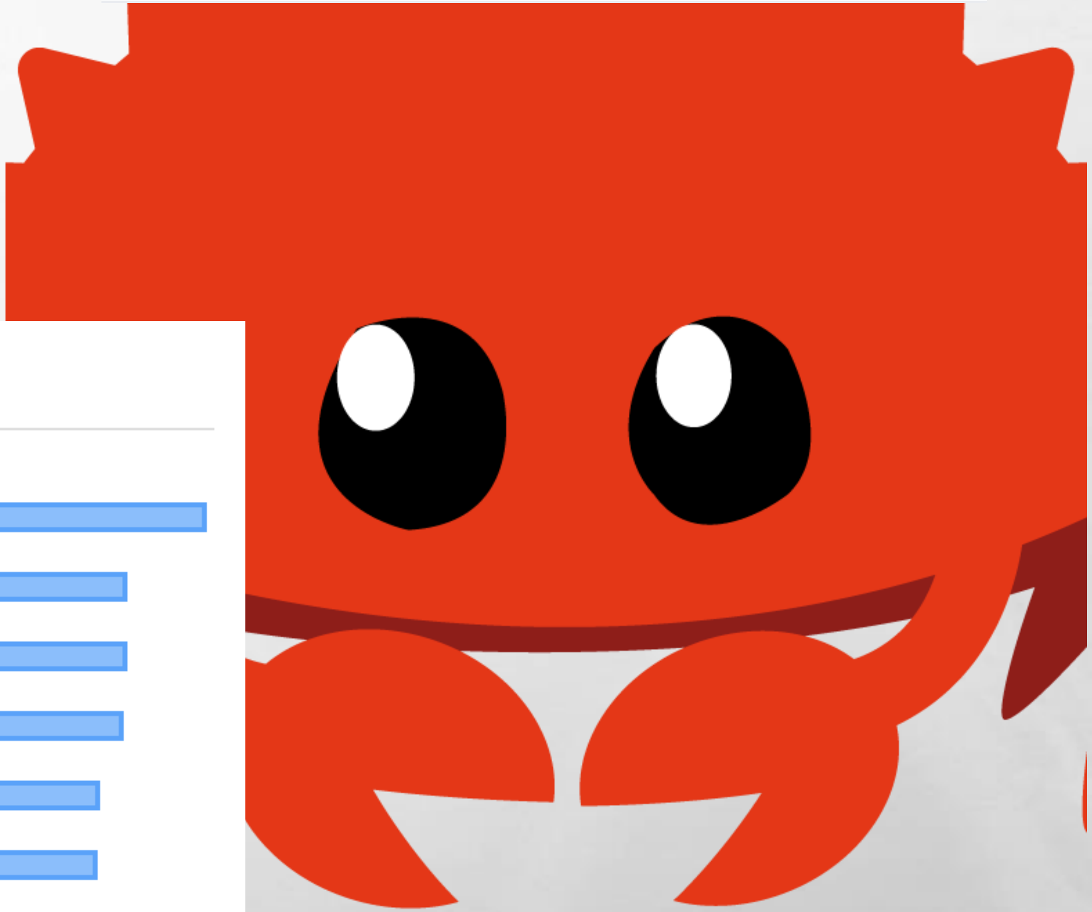
RUST? SHADERS?

RUST

- FAST AND MEMORY-SAFE SYSTEMS PROGRAMMING LANGUAGE

```
fn main() {
    println!("Hello, world!");
}
```





See more at: rust-lang.org

WHY RUST

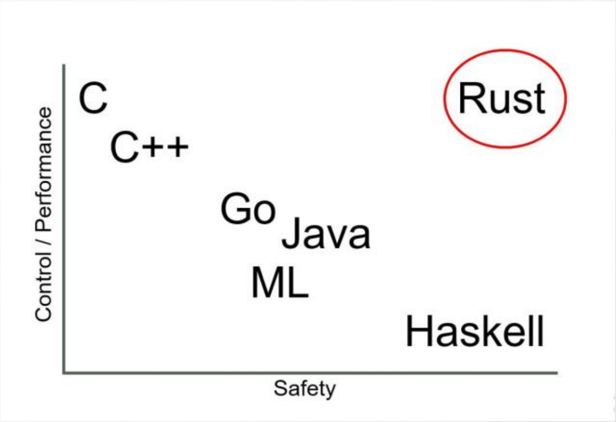
HERE'S HOW YOU ADD A CRATE

1. SEARCH ON CRATES.IO
2. COPY AND PASTE ONE LINE OF CODE:

```
[dependencies]
wgpu = "0.4.0"
```

CRATE == LIBRARY == DEPENDENCY == PACKAGE

USE == IMPORT



CPU VS GPU

CPU

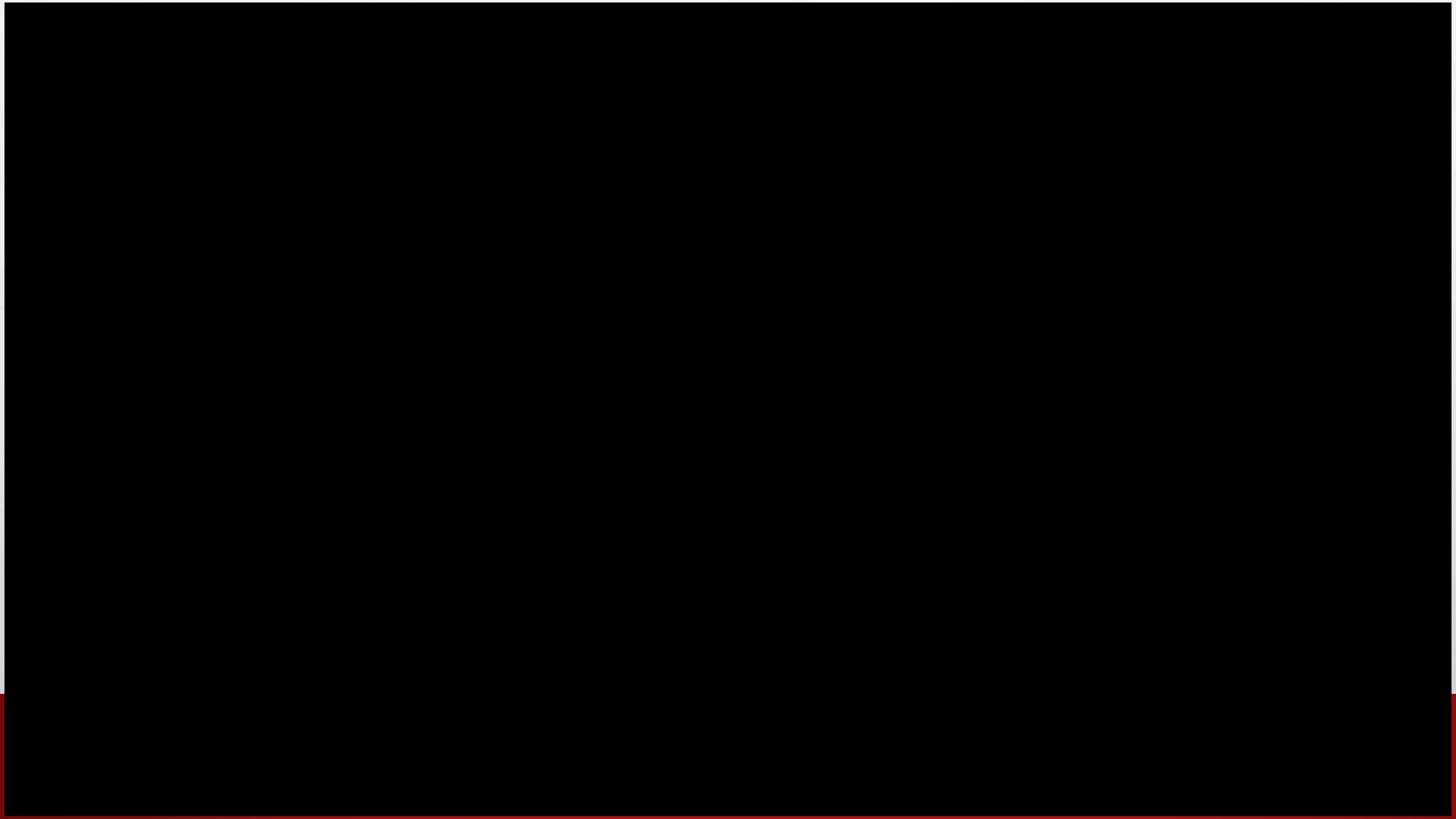
- FEW (16) HORSE SIZED DUCKS

GPU

- A LOT (2304)

OF DUCK SIZED HORSES

WHO BETTER TO EXPLAIN THIS THAN MYTHBUSTERS AND NVIDIA



SHADERS

IE. HOW TO TELL THE GPU TO DO THINGS

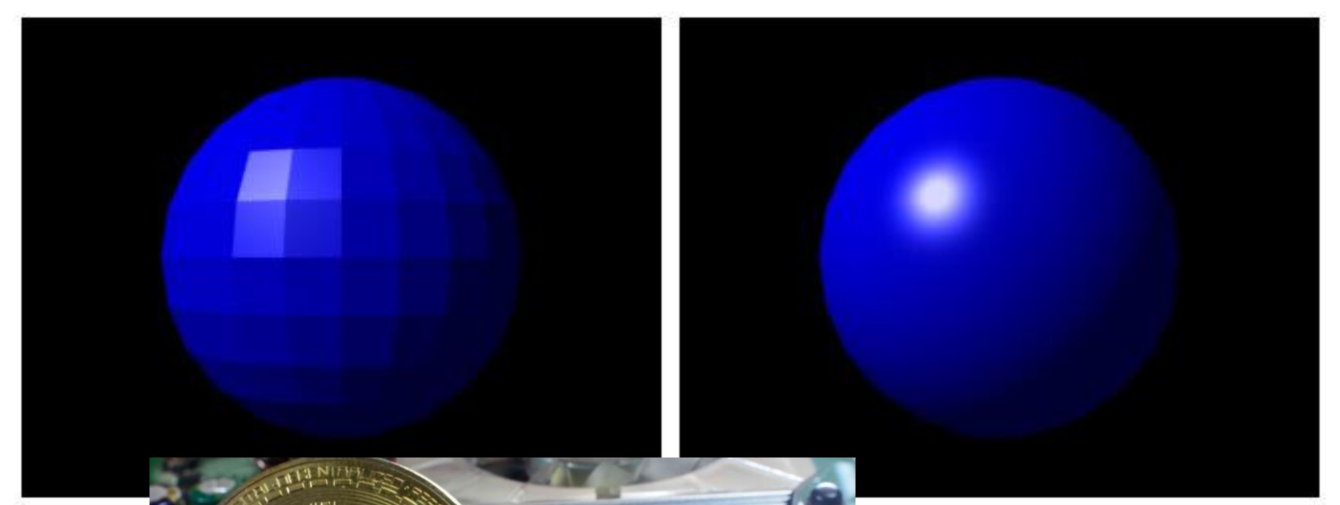
YOU NEED POWER. UNLIMITED POWAAH

Type of computer program

Very simple

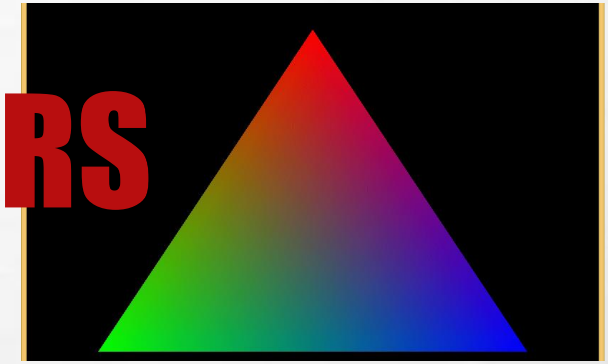
Designed to run in parallel

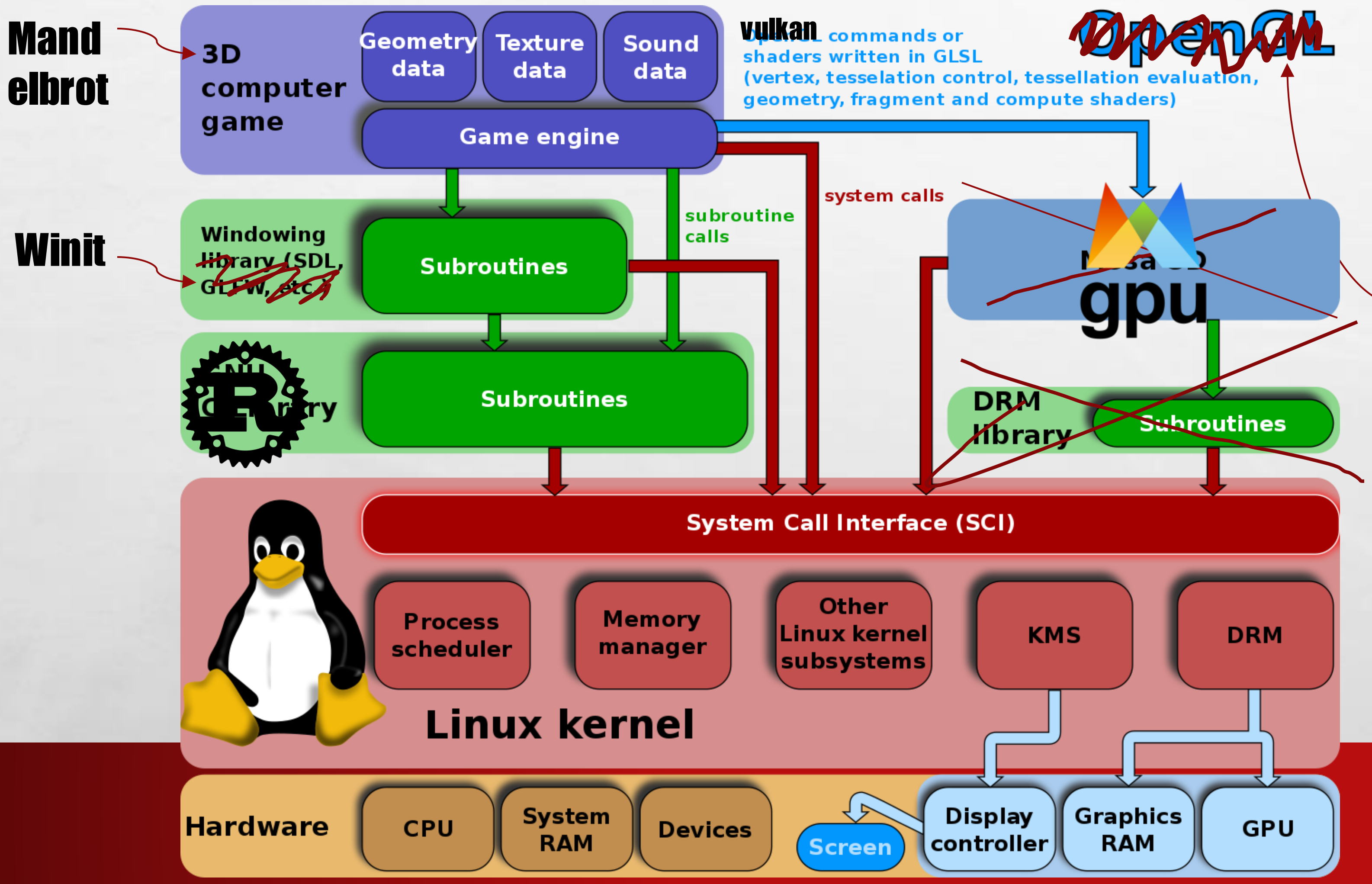
Many many times



PHONG SHADING

GLSL – HOW TO WRITE SHADERS





Vulkan™

SHADER CODE

GLSL

SHADER CODE

- THE SHADER CODE IS BASICALLY A FOR LOOP AND EVERY TIME IT ITERATES IT RETURNS DIFFERENT COLOUR THEN ADD THAT COLOUR TO POINTS THAT ESCAPED ON THAT ITERATION. IF A POINT DOES NOT ESCAPE THEN IT STAYS BLACK.
- SO ALL THE POINTS THAT ESCAPED AT THE SAME NUMBER OF ITERATION HAVE THE SAME COLOUR.

HOW IS IT CONVERTED INTO COLOURS

- FIRST OF ALL, WE NEED TO DEFINE HOW MANY ITERATIONS THERE IS GOING TO BE. THE MORE TIMES IT IS ITERATED THE MORE WE CAN ZOOM INTO IT.
- $Z = Z^2 + C$; // C IS A COMPLEX NUMBER.
- IF C ESCAPES TO INFINITY THAT POINT IS COLOURED BASED ON HOW QUICKLY IT ESCAPES. THAT'S HOW WE GET NIFTY SHADES. IF IT IS BOUNDED THEN IT IS COLOURED BLACK.
- IN OTHER WORDS, IF THE ABSOLUTE VALUE OF Z ON N^{TH} ITERATION NEVER BECOMES LARGER THAN A CERTAIN NUMBER (THAT NUMBER DEPENDS ON C), NO MATTER HOW LARGE N GETS, THEN IT IS COLOURED BLACK.

HOW IS IT CONVERTED INTO COLOURS

- FOR EXAMPLE, IF C = 1 THEN THE SEQUENCE IS 1, 2, 5, 26,... WHICH GOES TO INFINITY. THEREFORE, 1 IS NOT AN ELEMENT OF THE MANDELBROT SET, AND THUS IS NOT COLOURED BLACK.
- $F(0) = 0^2 + 1$ (1)
- $F(1) = 1^2 + 1$ (2)
- $F(2) = 2^2 + 1$ (5)
- $F(5) = 5^2 + 1$ (26)

HOW IS IT CONVERTED INTO COLOURS

- LET'S TRY IT WITH Z = 0
- F(0) = 0^2 + -1 (-1)
- F(-1) = -1^2 + -1 (0)
- F(0) = 0^2 + -1 (-1)
- AS SEEN ABOVE 0 IS BOUNDED BETWEEN -1 AND 0 SO IT IS IN MANDELBROT SET AND COLOURED BLACK.

HOW IS IT CONVERTED INTO COLOURS

- TO DETERMINE WHETHER C IS IN MANDELBROT SET OR NOT, WE ITERATE 0 UNDER SQUARE(Z) + C
- IF IT GETS BIG QUICKLY THEN WE GIVE IT A COLOUR. IF IT TAKES LONG TIME TO GET BIG WE GIVE IT A DIFFERENT COLOUR. IF IT TAKES EVEN LONGER WE GIVE IT ANOTHER COLOUR.

JULIA SET

- IT IS IMPORTANT TO UNDERSTAND THAT THE ONLY THING THAT IS DIFFERENT IN DIFFERENT IMAGES OF JULIA SETS IS THE NUMBER C. CONSIDERING THAT THERE ARE INFINITE AMOUNT OF COMPLEX NUMBER C, THERE ARE INFINITE AMOUNT OF JULIA SETS.

JULIA SET

- THERE ARE TWO DIFFERENT GROUPS IN EVERY JULIA SETS.
- FIRST GROUP OF JULIA SETS CONTAINS BLACK REGIONS OF COMPLEX NUMBERS THAT STAY BOUNDED. IT IS CALLED CONNECTED JULIA SETS BECAUSE THEY CONNECT TOGETHER AND FORM A REGION.
- THE OTHER GROUP JULIA SETS CONSIST OF NON-BLACK DISCONNECTED POINTS.
- IF 0 ESCAPES TO INFINITY THEN IT MEANS THAT EVERY Z NUMBER ESCAPES AND THERE IS NO BLACK REGION WHATSOEVER IN THAT JULIA SET. IF 0 STAYS BOUNDED THEN IT THERE IS A BLACK REGION AND 0 IS A PART OF THE REGION.
- $F(Z) = Z^2 + C$ // GIVEN THAT $Z = 0$ WE CHANGE C AND DETERMINE IF IT IS CONNECTED OR DISCONNECTED.

MANDELBROT SET

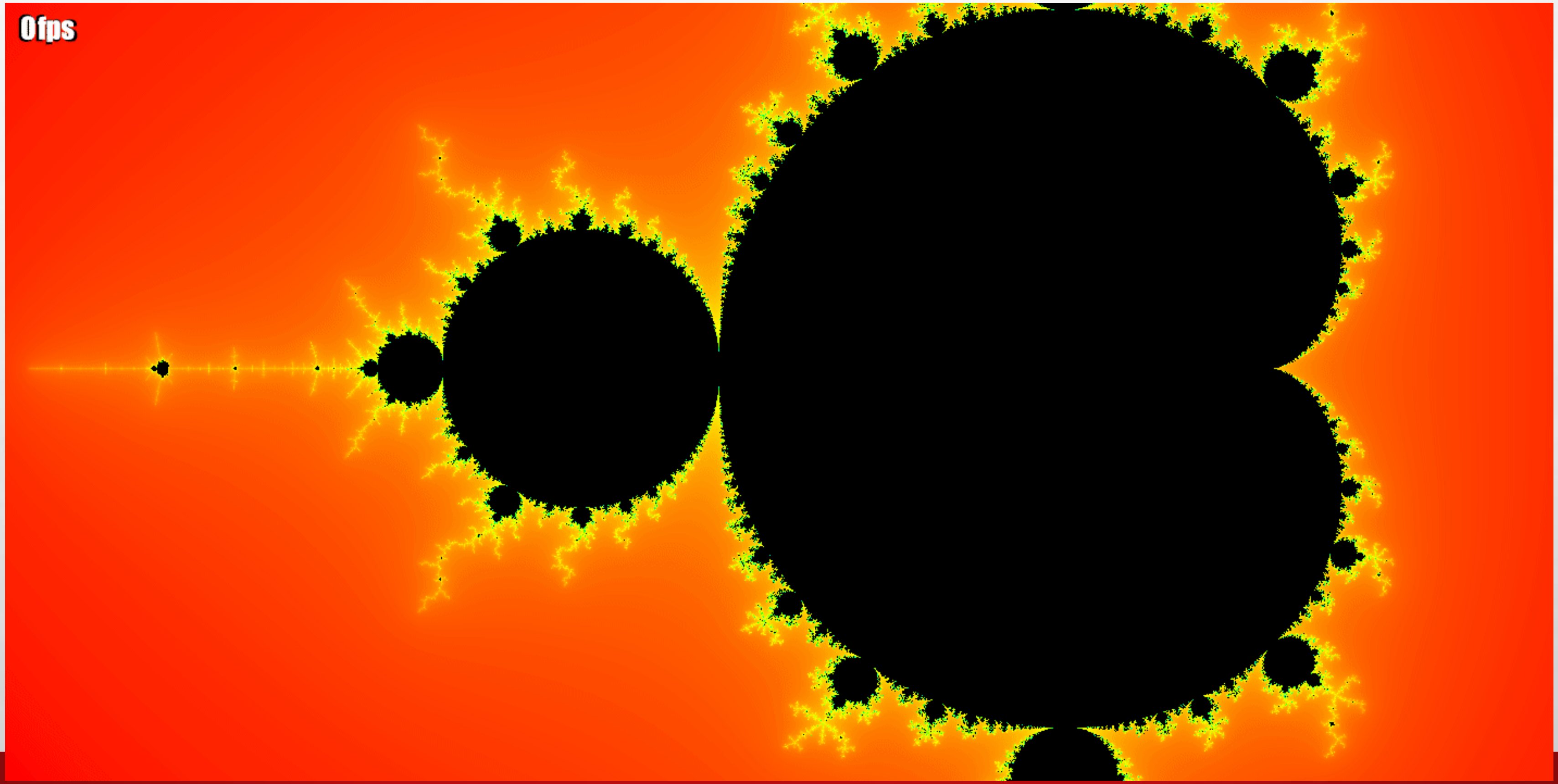
- EVERYTHING IN MANDELBROT SET IS WITHIN THE DISTANCE 2 OF THE CENTRE. ONCE A NUMBER ESCAPES, IT IS OUT OF THE PICTURE THUS NOT COLOURED BLACK.

FEATURES

MORE THAN 0, (IN ORDER OF IMPORTANCE)

1. CJ CAMEO

ZOOM, PAN AND CHANGE NUMBER OF ITERATIONS INTERACTIVELY



(KIND OF) WORKING FPS COUNTER

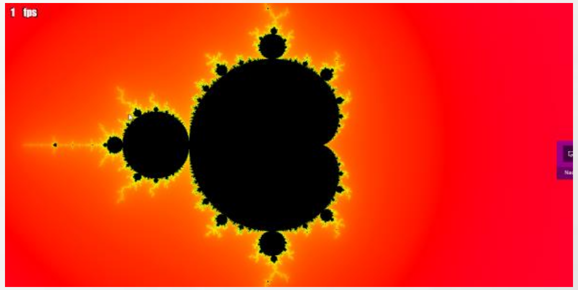
SHADER LIVE RELOAD

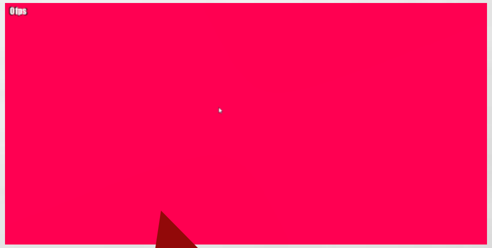
WHENEVER YOU CHANGE THE SHADER IT AUTOMATICALLY RELOADS.

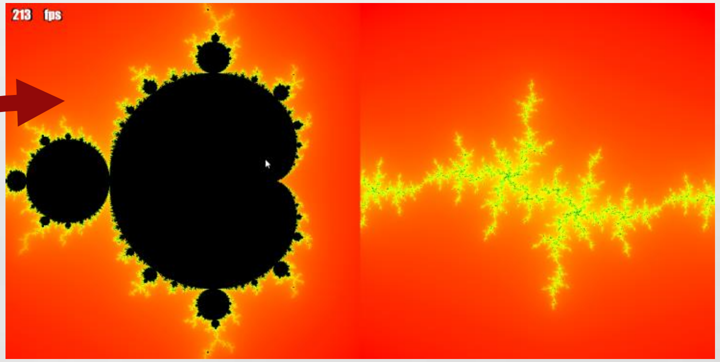
PLEASE DON'T WRITE BAD CODE OR IT'S GOING TO CRASH THOUGH!

MULTIPLE TYPES OF VIEWS

- 1 – JUST MANDELBROT
- 2 – JUST JULIA (PRETTY BORING)
- 3 – MANDELBROT AND JULIA (NOT FINISHED BUT PROOF OF CONCEPT)







Not working correctly
(but looks cool!)

Boring
Julia with big constant

DEMO