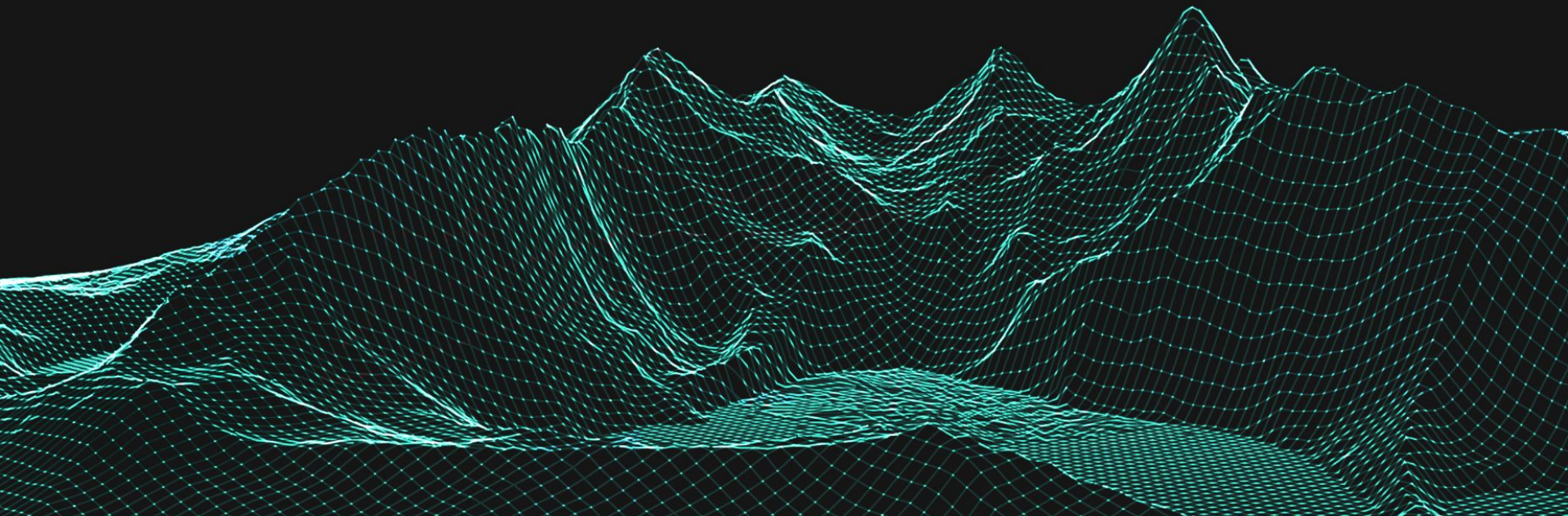
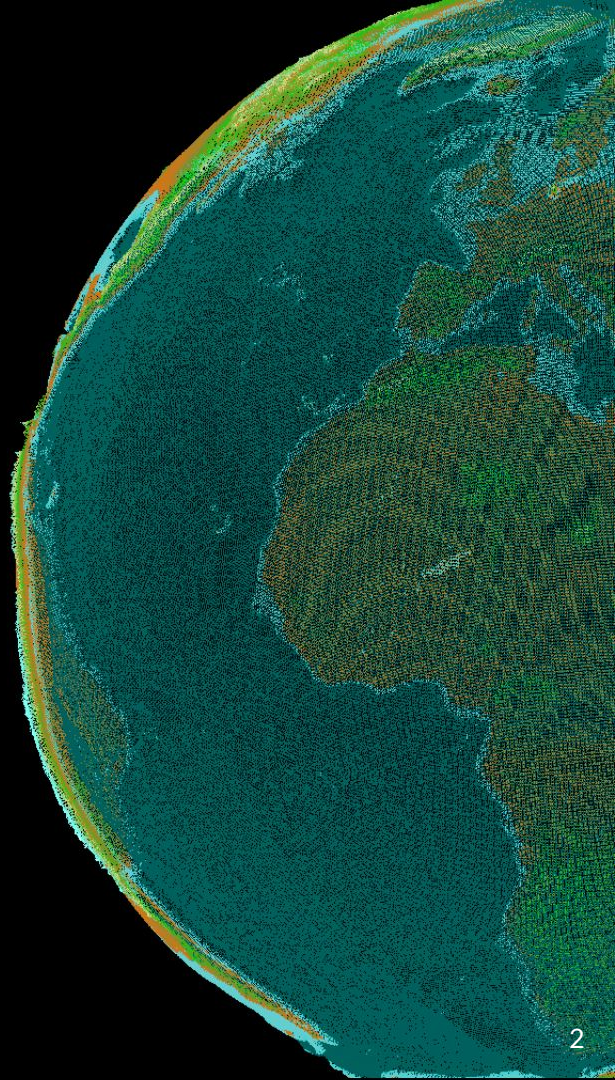
FdF

A graphic programming project



Who are we?



Jorge Cruz

- Mechanical Engineer from FEUP.
- Masters in Automation
- 5+ years of experience in mechanical design for production

Rui Pires

- Bachelor in Classical Music from Hochschule für Musik Karlsruhe - Germany
- 6+ years of experience in Classical and Wind Orchestra

Project objective

Create a wireframe model representation of a 3D landscape by linking various points with line segments (edges).

FdF : Fil de Fer

Content

Project Overview

- Overview
- Problems to solve
- Solution Summary

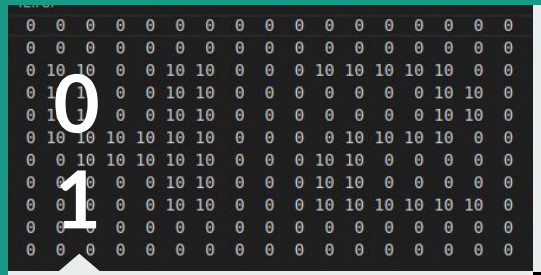
Overview

- Graphics programming
  - How to place points in space
  - Algorithms to join points in space
- Trigonometry and projection views
  - Rotation Matrices
  - 3D points to a 2D view

Problems to solve

1. Read numbers from a file and convert into usable data.
2. Create and manage a new window using minilibx.
3. From the usable data, make calculations to represent each point on screen.
4. Implement an algorithm to paint pixels between points to create line segments.

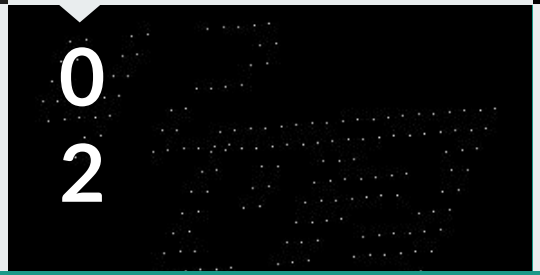
Solution summary



01

Read from input file

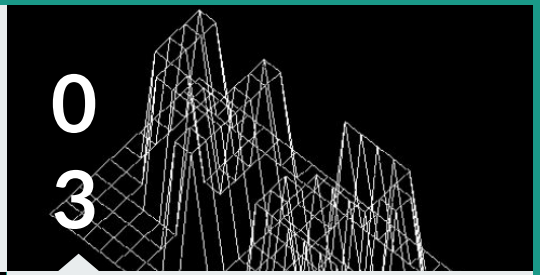
Create an array of points from a document



02

Going 3D

Equate the points of the array into isometric perspective



03

Connect the dots

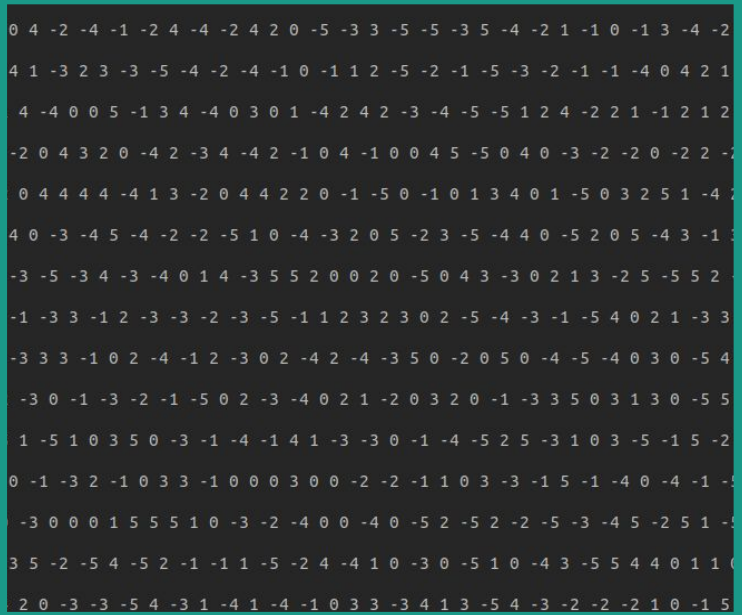
Implement an algorithm to connect two consecutive points

Technical Details

- Where to start?
- Projection Calculation
- Linking points through segments
- Results

Where to start?

- Create a matrix considering the data read
- Fill the matrix with data from the document



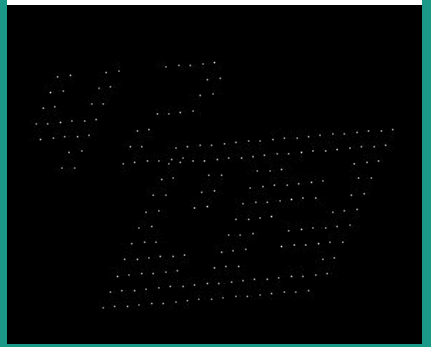
Projection Calculation

- Use rotation matrices to adjust to an isometric projection
- Project the final points to the viewport screen

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

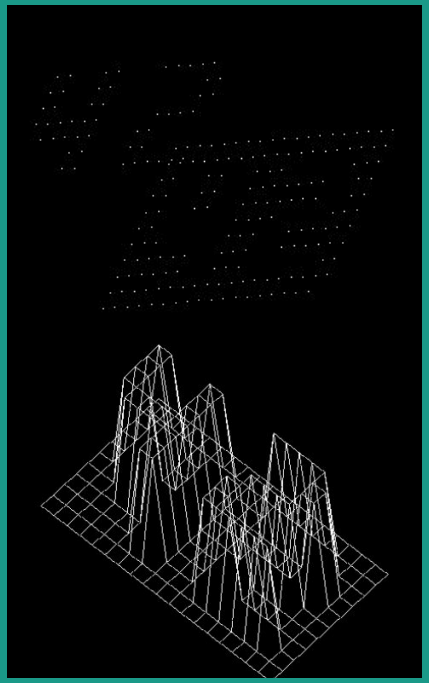
$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

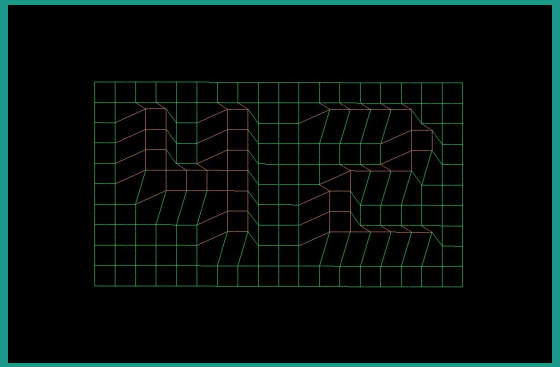


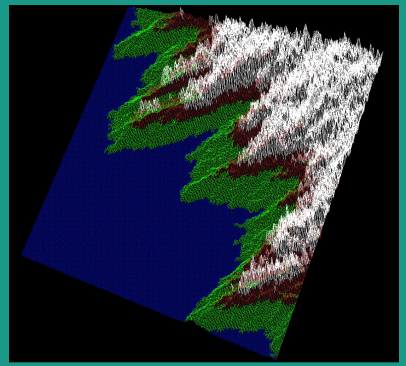
Linking points

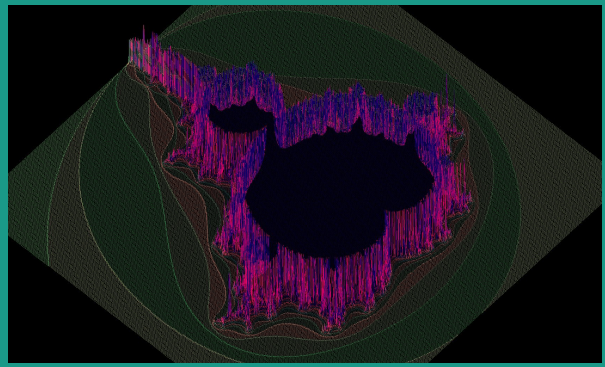
- Define a step dimension to colour pixels
- Apply a selection algorithm to paint the correct pixel



First Results





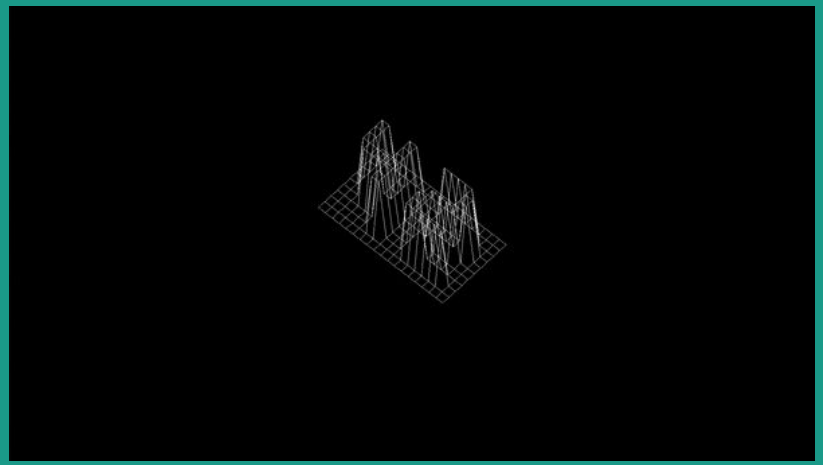


Bonus

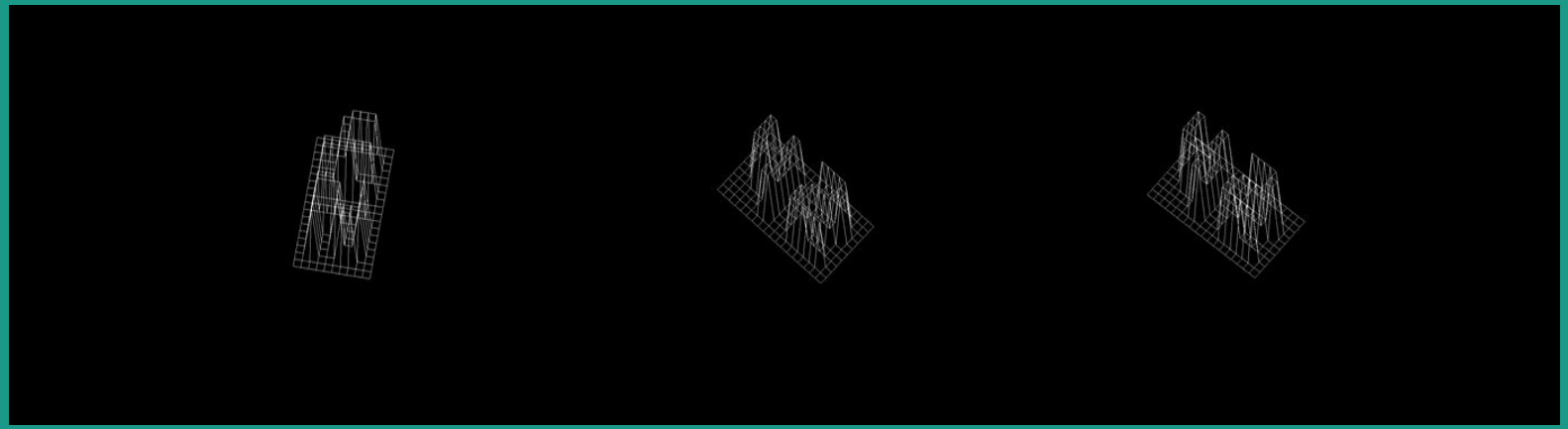
Content

- Additional Projections
- Zoom
- Rotation of the model
- Translation of the model
- Implement a different set of colours

New set of colors



Rotation of the model



Zoom and Translation of the model

Why continue?

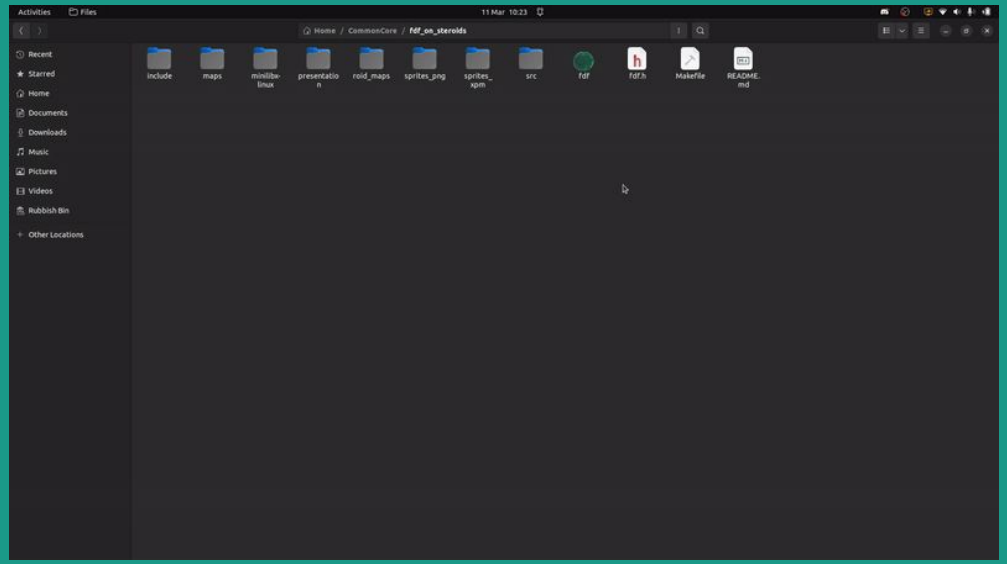
- It is an opportunity to work with peers
- Testing what we can do with a simple library (minilibx) for graphic programming
- At 42 we are encouraged to overcome barriers and to experiment

(Because it looks cool…)

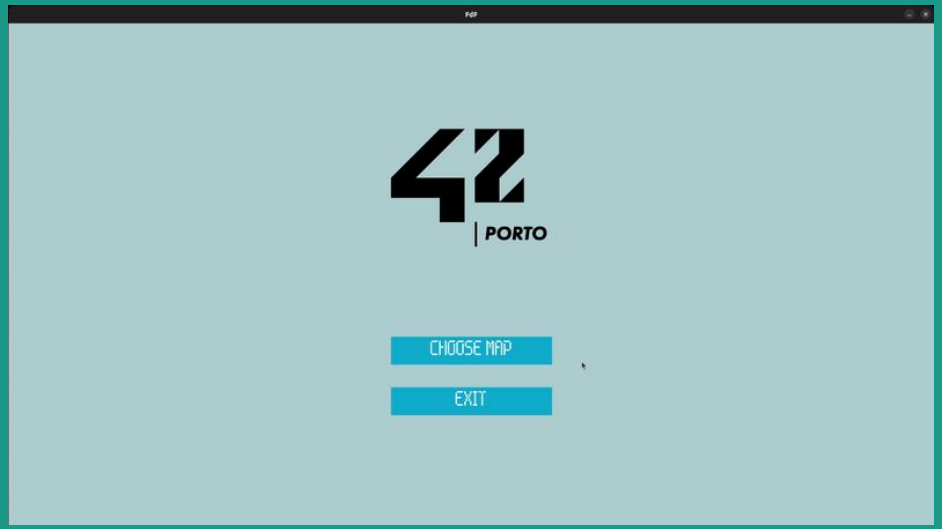
Additional Content

- Stand-alone app
- Selector screen
- Control panel
- Color schemes
- Manual adjustments
- Additional projections

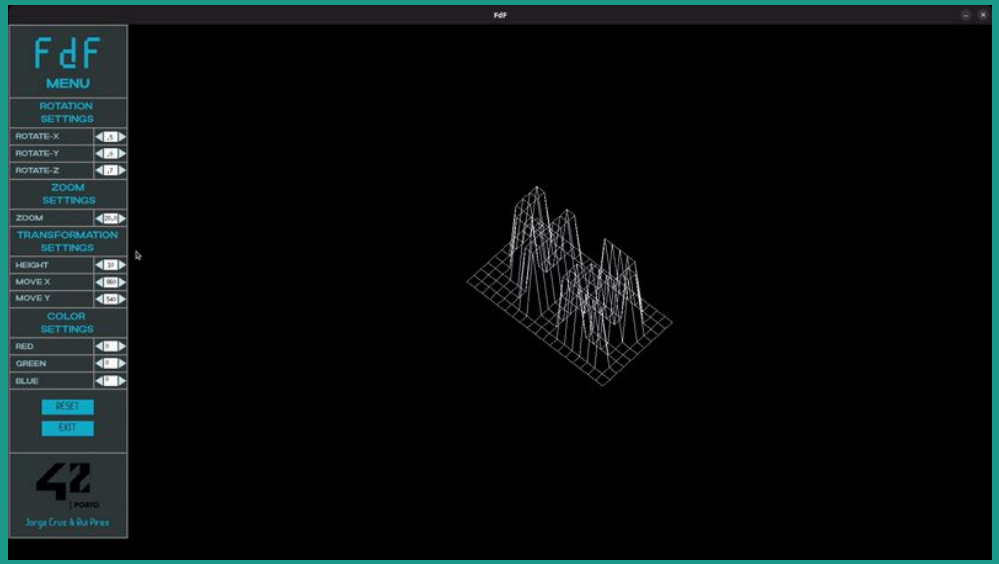
Stand-alone application



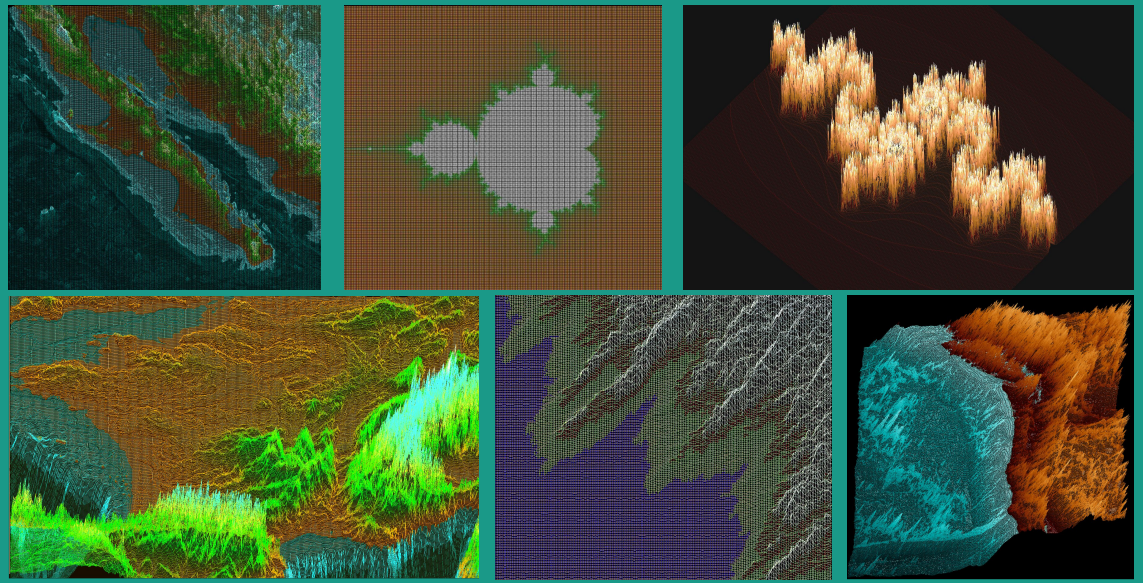
Selector screen



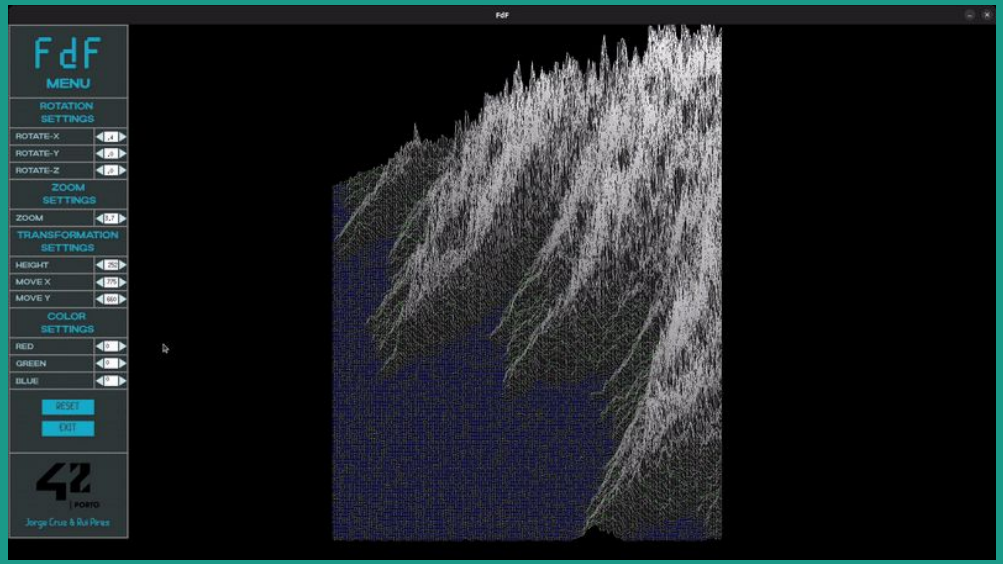
Control Panel



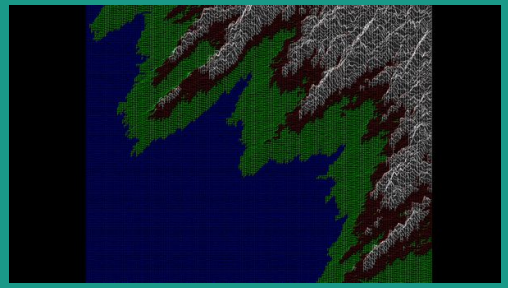
Improved colour implementation

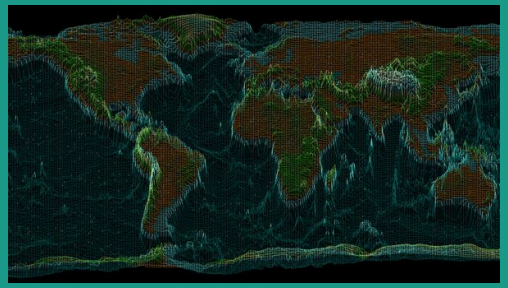


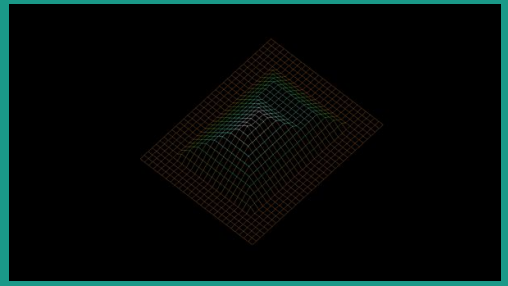
Adjustment of colours on execution

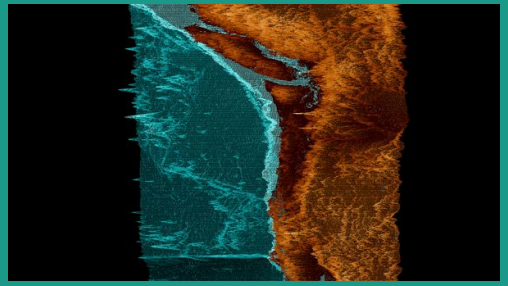


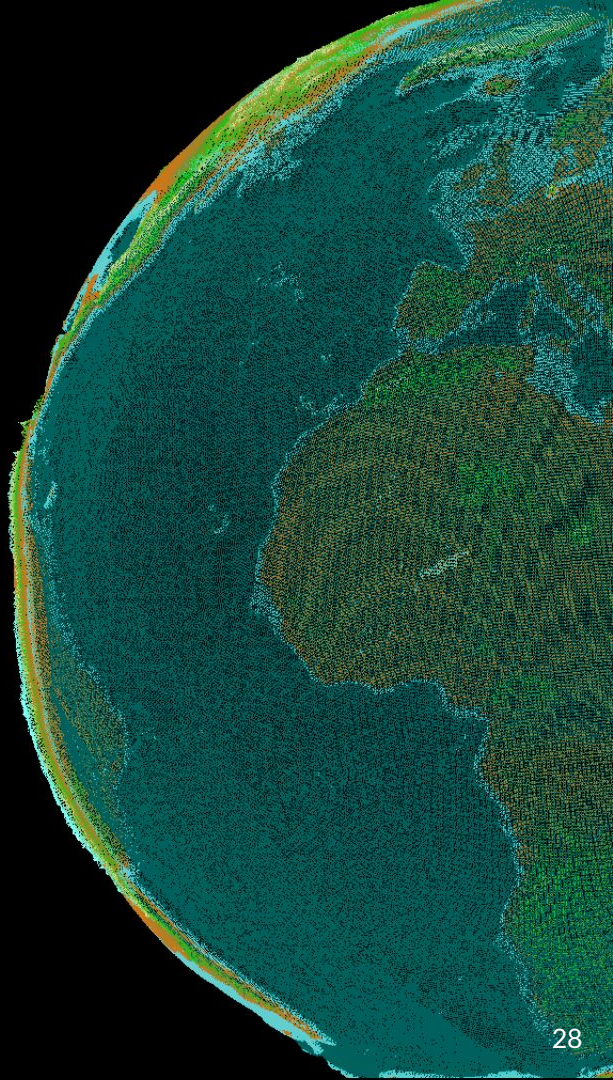
Projections with a twist





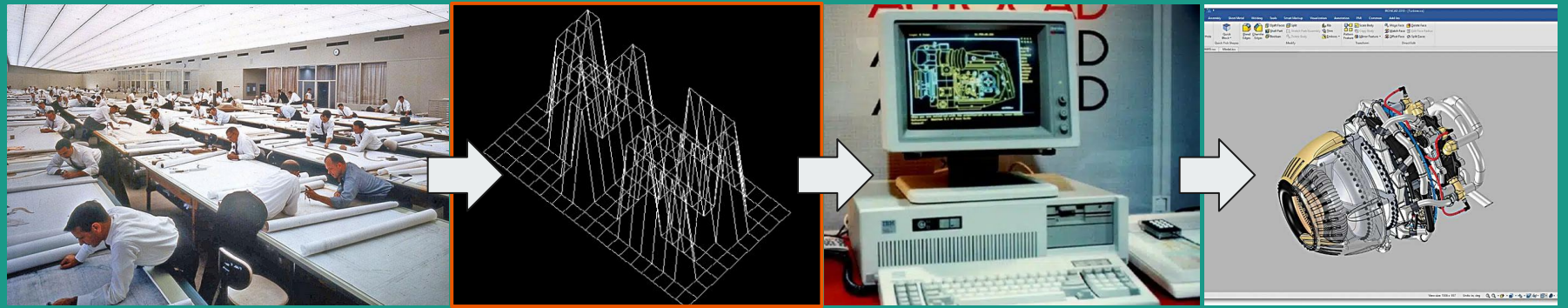






Applications

- This is a learning project in which we start from scratch;
- Understand where 3D file renderers and CAD softwares start from.



Before 1970s | 1970s | 1980s | 2020s

Thank you