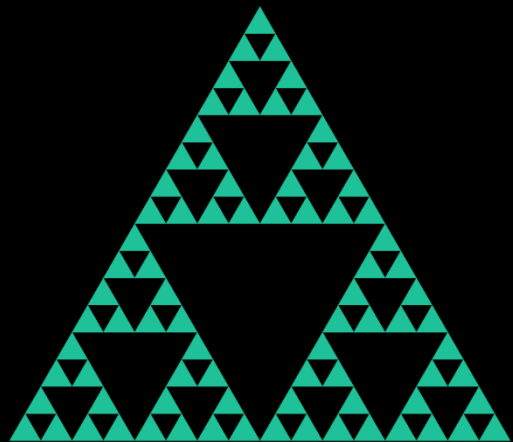


Fractals

Web app



Ana Gloria Galvez Mellado

MUNI-SPRING 2021 Visual Creativity Informatics

Index

Fractals 2

The topic: Iterated function systems 2

User interface 3

 General settings 3

 Specific settings 4

 Tree settings 4

 Koch settings 5

 Barnsley fern settings 6

 Sierpiński settings 7

Nice results in short time 8

More examples 9

Fractals

Fractals is a browser app based on iterated function systems (IFSs). It contains some of the most-famous examples of these fractals, and some experimental variations of them.

The app allows the user to change different parameters (based on fractal type), save them as a PNG with several resolutions and save/load the project settings in order to recreate them later.

The idea of the app is to compile those examples with educational and illustrative purposes – and a really intuitive GUI –, such as quickly show how different changes (length, angle, iterations) can affect them. No mathematical level is required, which means that it can be a great web app for kids/teenagers and schools.

The topic: Iterated function systems

Iterated function systems (IFSs) are a mathematical construct used to easily represent sets of fractals that are made up of the union of several copies of itself, each copy being transformed by a function (hence "function system").

The functions are normally contractive, which means they bring points closer together and make shapes smaller (see Tree, Koch, ...).

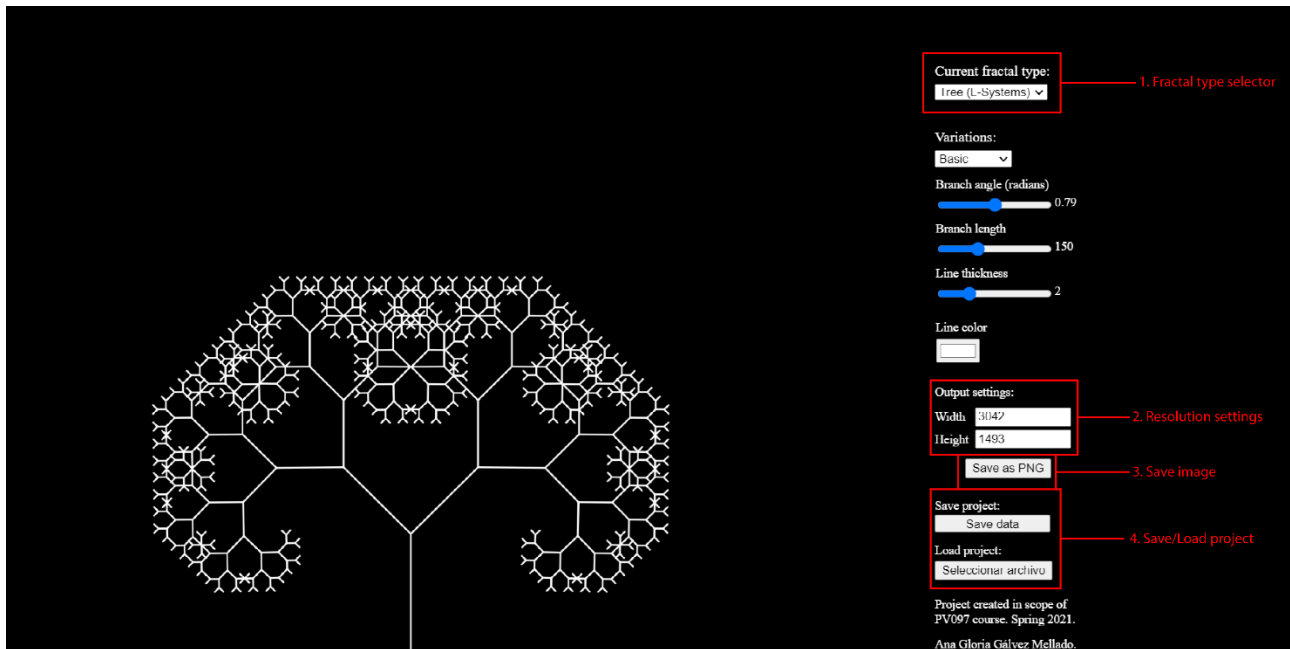
The shape of an IFS fractal is made up of several possibly-overlapping smaller copies of itself, each of which is also made up of copies of itself, ad infinitum. This is the source of its self-similar fractal nature.

It also can be done by using random iteration algorithms (also called "chaos game"), which consist of selecting a random point and create a fractal by iterating around the original point.

User interface

The settings panel is located on the left side of the screen. On it, we can see settings that are common for all fractals (General settings), and some **Specific settings**, depending on the fractal selection. Their functionality is described below:

General settings



1. **Fractal type selector:** It's used to change the type of fractal (Tree, Koch, Barnsley fern, Sierpiński).
2. **Resolution settings:** In here, the output width and height (on pixels) of the image can be changed. Consider that the drawing is re-drawn each time the resolution is changed.
The aspect ratio is fixed (based on the current window ratio), so if the width value is changed, the height will be automatically recalculated, and vice-versa.
Max. resolution allowed is 8K.
3. **Save image button:** Once pressed, it will store the current image with given resolution in PNG format.
4. **Save/Load project:** "Save data" button will store the information of the current image on a .JSON file.

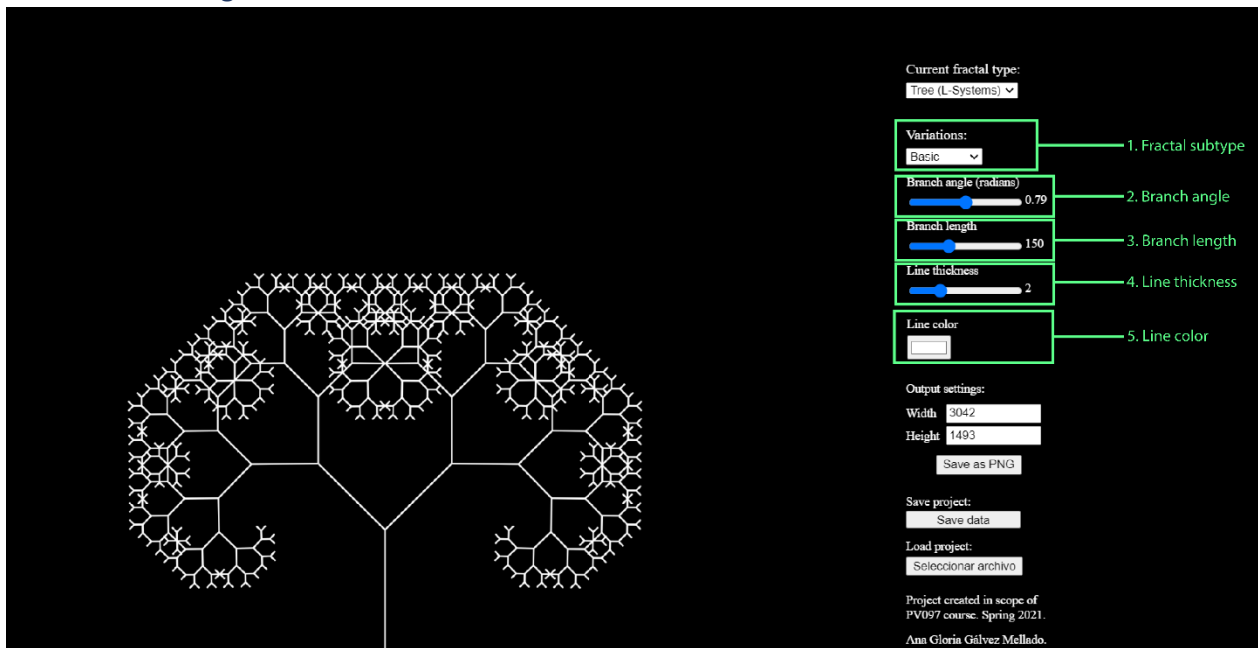
"Load project" button will ask for a JSON file project with settings, and automatically load the parameters and draw the image.

If the file uploaded has not the proper format, nothing would happen.

Specific settings

Each of the different fractals has their own specific settings for changing their parameters.

Tree settings



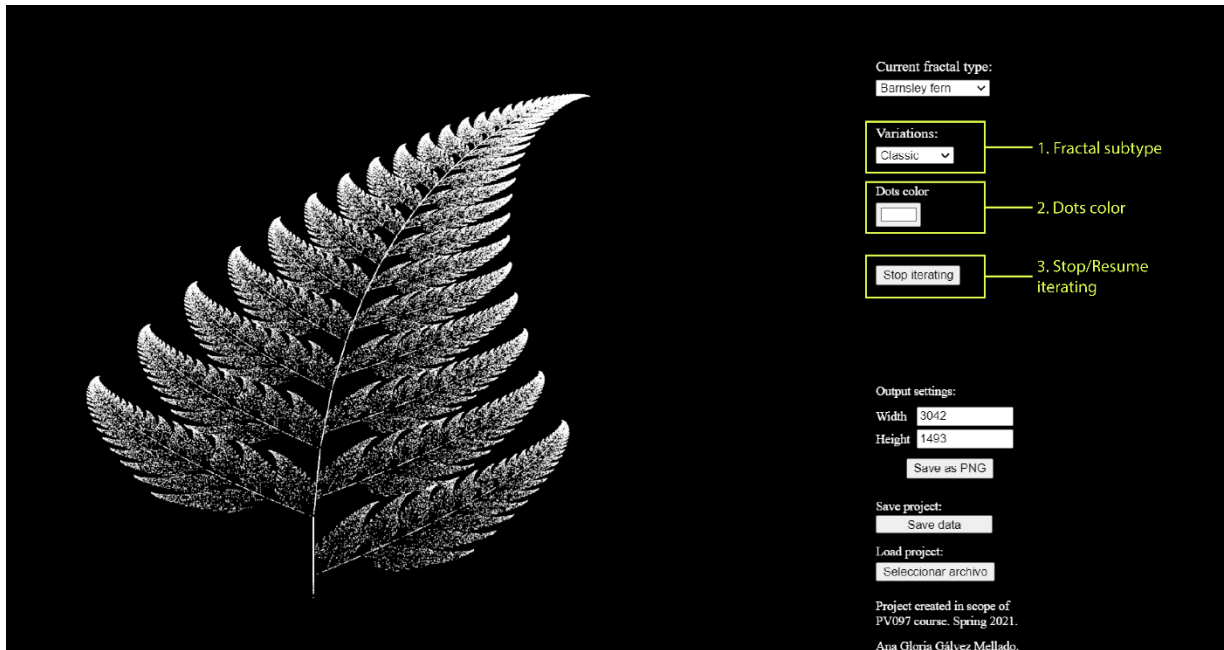
1. **Fractal subtype:** It's used to alternate between the different variations of the tree.
2. **Branch angle:** This parameter adjusts the angle (radians) between the branches of the tree.
3. **Branch length:** It can be used to regulate the length of the branches of the tree (it changes the size of the longest one, the rest have proportional sizes).
4. **Line thickness:** To adjust how thick is the line of the drawing.
5. **Line color:** To change the color of the tree.

Koch settings



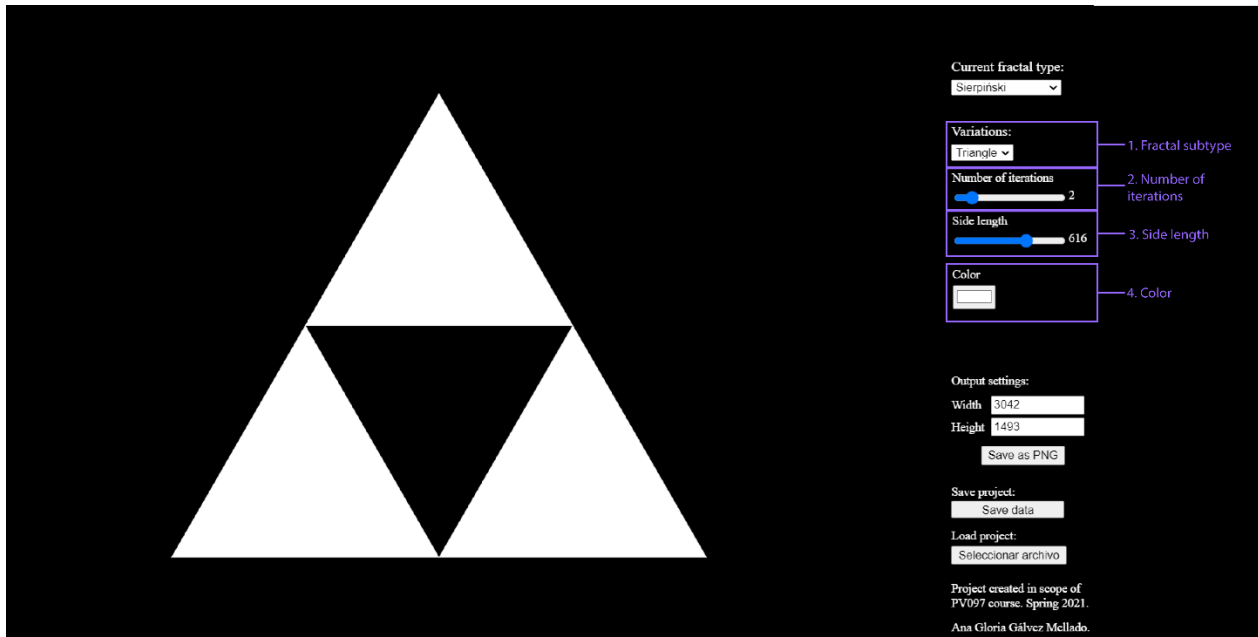
1. **Fractal subtype:** It's used to alternate between the different variations of the fractal (Koch line, Koch snowflake, Koch antisnowflake, Minkowski sausage, Minkowski island).
2. **Number of iterations:** To regulate the amount of iterations for this fractal (the more iterations, the more complex the shape, but also it will last more time to process the image).
3. **Line length:** To adjust the length of the main lines (the others are proportional to the biggest ones).
4. **Line thickness:** To adjust how thick is the line of the drawing.
5. **Line color:** To change the color of the fractal.

Barnsley fern settings



1. **Fractal subtype:** It's used to alternate between the different variations of the fern.
2. **Dots color:** To change the color of the dots that will be painted on the next iterations. Note that the color of the previous painted points is not changed, so an effect of several colors for the same drawing can be achieved by changing these parameters several times.
3. **Stop/Resume iterating button:** When pressed, no more points will be drawn at a time. When pressed again ("Resume iterating"), it will continue drawing points.

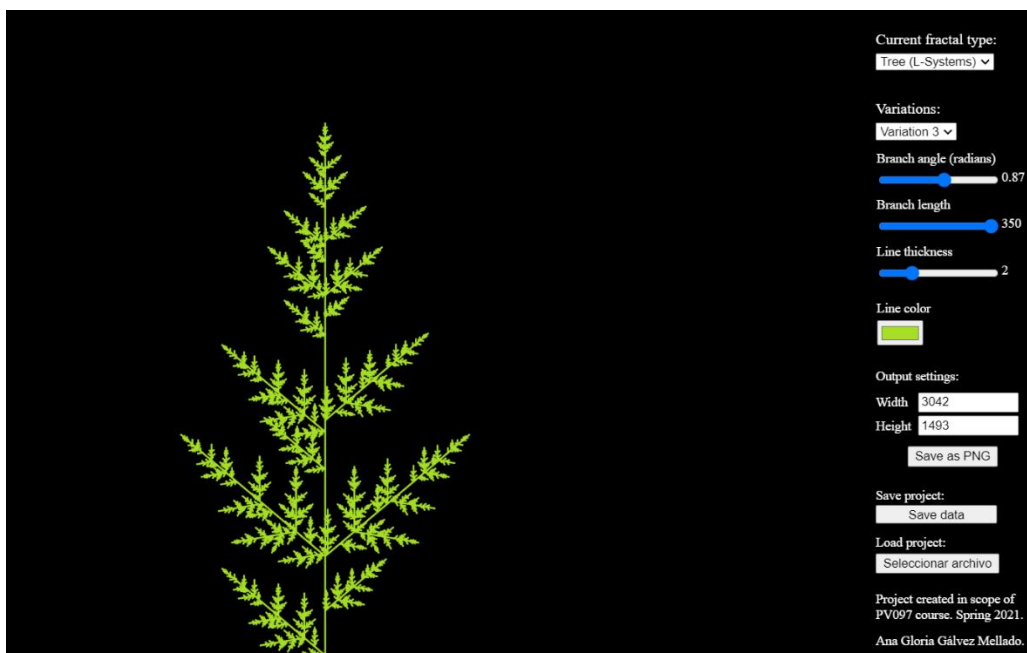
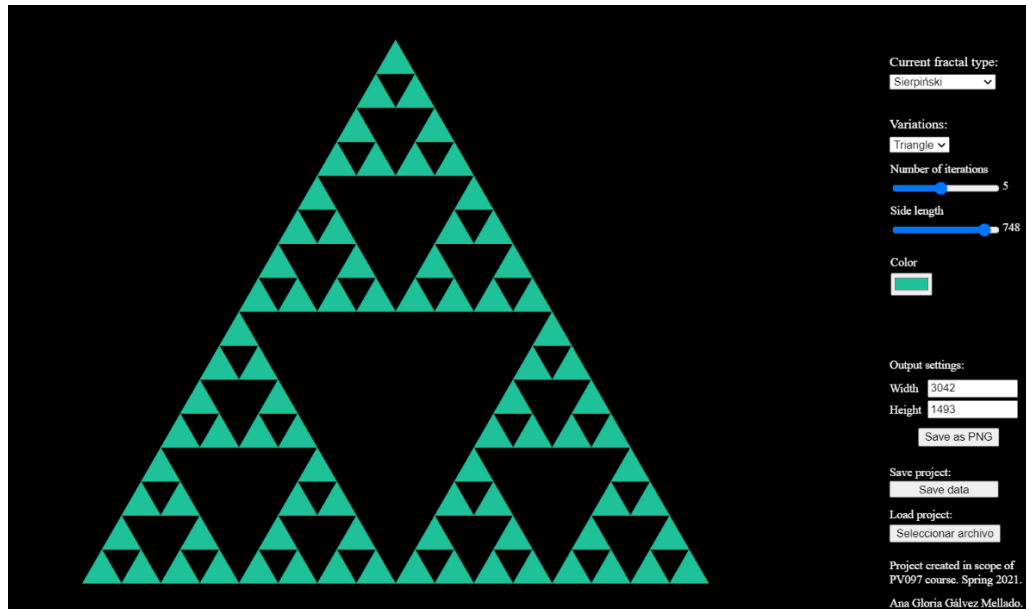
Sierpiński settings

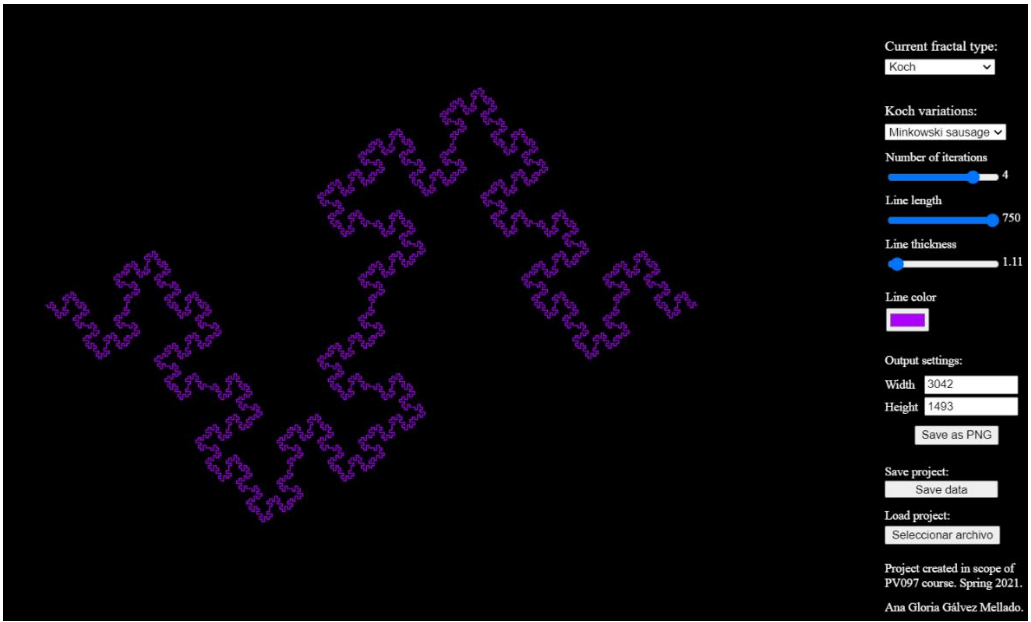


1. **Fractal subtype:** It's used to alternate between the different variations of the fractal (triangle, carpet).
2. **Number of iterations:** To regulate the amount of iterations for this fractal (the more iterations, the more complex the shape, but also it will last more time to process the image).
3. **Side length:** To adjust the length of the main figure (the others are proportional to the biggest ones).
4. **Color:** To change the color of the fractal.

Nice results in short time

Obtaining nice results with *Fractals* is as easy as open the app and start changing parameters. Any of the result will be nice as all the fractals are beautiful by themselves. Here are some screenshot examples (so the parameters used can be seen):





More examples

