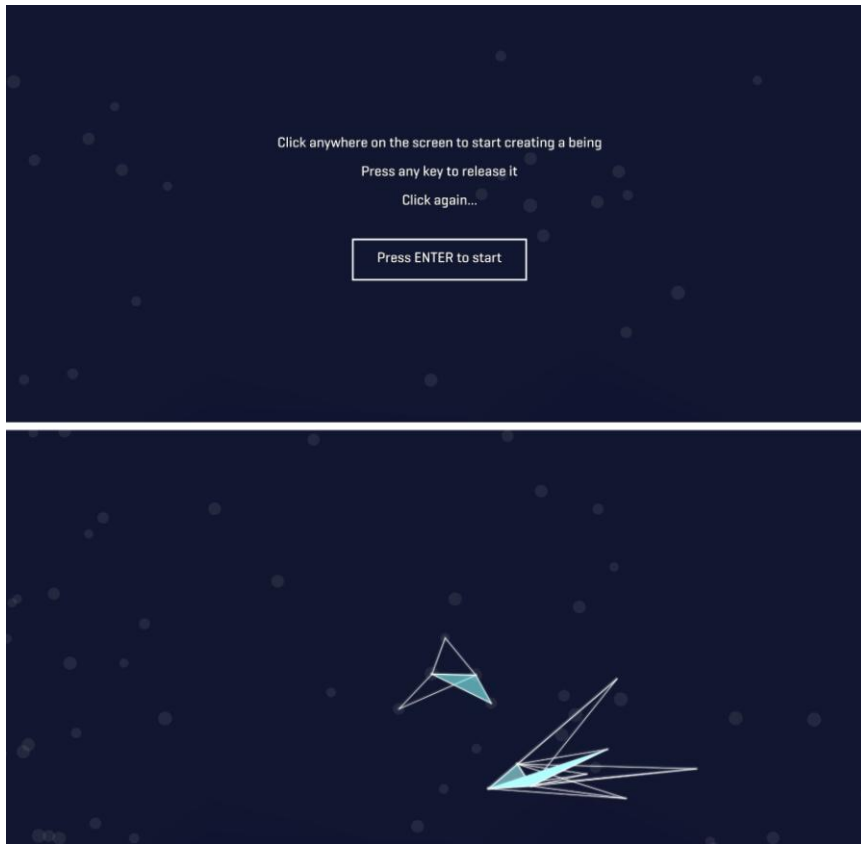


FLOAT



Introduction

I find extremely exciting to be able to build entire interactive worlds from scratch, and have long played with [Processing](#) to build small generative sketches.

I wanted to test its browser counterpart, [P5js](#), and decided to create a small game.

A hosted version of **Float** can be found [here](#). The code can be viewed [here](#).

The App

Float is a game in which users generate creatures. Once released, the creatures live their own life around the screen.

How to use the site

Visitors can click anywhere on the screen, creating a small bubble, a semi-transparent circle. Each of the circles created is linked to the others with a line, creating a small creature. Once released, the creature is free to move by itself, gently pushed by the water flow, and forming flocks with other creatures floating around the screen.

How it was built

Float is a JavaScript application, built with [P5js](#) library.

The physic engine that controls its creatures' movements is based on the tutorials from the genial [Daniel Shiffman](#).

Challenges during the build

While I had a fairly good idea of what kind of look and feel I wanted to give to the game, I realised through its development that its partially random nature meant that the 'ideal' designs I had in mind initially were not always possible.

For instance, connecting all the points of a creature to all of its other points creates a very busy design that isn't very nice. Instead, it is much better looking when only 2 points are connected to all others. In a fixed environment (ie. set of pre-generated creatures), I would have been able to decide which of the cloud of points these two points would be.

But as the creatures are generated by users, the two points have to be selected 'randomly', leading to a less controlled design. I realised after going back to the drawing board that it was marginally nicer to have the two closest points being the two 'connected' points. However the implementation of that feature considerably slowed down the game processing, leading to a bad experience, so I abandoned it.

Coding for generative design is an interesting process; it reminds me of my own craft / artistic process with physical making, where the type of material and the will of the artist combine to create the artwork.

Credits

Physic engine based on [Daniel Shiffman](#)'s tutorials.