

Programma di massima. (**BOZZA**)

0.0. Comprendere e fare pratica delle tecniche di base della programmazione come esposte e illustrate in https://www.mariochiari.net/TO2526/risorse/Learning_Processing.pdf

0.1. Se non lo avete ancora svolto, finire il vostro Personaggio, fino alla programmazione di una moltitudine di personaggi mediante definizione di opportuna classe.

1. Ricorsività, Frattali e L-system

<https://natureofcode.com/fractals/> Cap. 8

<https://thecodingtrain.com/challenges/16-l-system-fractal-trees>

<https://thecodingtrain.com/challenges/174-graphics-applesoft-basic>

2. Flock and Boids

<https://thecodingtrain.com/challenges/124-flocking-simulation>

3. Perlin Noise

Vedi Track NOISE a

<https://thecodingtrain.com/tracks/noise/noc/perlin/intro-to-perlin-noise>, e i video seguenti (sono nove)

4. Machine learning

<https://www.youtube.com/watch?v=3MqJzMvHE3E> Challenge 158

<https://natureofcode.com/neural-networks/> (da valutare quanto sia adatto)

Per ognuno dei quattro argomenti:

1. saper spiegare di che cosa si tratta, in linea di massima, ma senza errori e errate concezioni;
2. saper riprodurre gli esempi di Shiffman in locale sulla vostra macchina;
3. **saper modificare gli esempi, per quanto possibile;**
4. curiosare in giro (google, risorse consigliate, altri tutoriali, chatGTP,), e vedere come ognuna di queste tecniche possa essere utilizzata per qualcosa di più advanced (senza perdersi).

Buon lavoro.

Risorse utili

Flock and Boids

<https://www.red3d.com/cwr/boids/>

di Craig Reynolds, che ha sviluppato l'algoritmo <https://www.red3d.com/cwr/index.html>

(i link a una varietà di risorse sono però in gran parte dead)

<https://dn710108.ca.archive.org/0/items/boids-demo-reel-1/Boids%20Demo%20Reel%201.mp4>

(NICE!)

<https://en.wikipedia.org/wiki/Boids>

e inoltre su youtube.com trovate decine di esempi di animazioni, cercate con: Flocking Boids

Perlin Noise

Perlin's web pages <https://cs.nyu.edu/~perlin/>

-- paper <https://mrl.cs.nyu.edu/~perlin/paper445.pdf>

(il secondo suo paper tecnico, con miglitorie, difficile)

-- programma <https://cs.nyu.edu/~perlin/noise/> (da capire)

<https://adrianb.io/2014/08/09/perlinnoise.html>

(noto tutorial introduttivo, ma in parte obsoleto, links dead) (by Adrian Biagioli)

<https://thebookofshaders.com/11/> (da *The Book of Shaders* by Patricio Gonzalez Vivo & Jen Lowe).

(forse anche più chiaro)

(sarebbe utile riscrivere il codice in Processing)

<https://sighack.com/post/getting-creative-with-perlin-noise-fields>

(esempi di applicazione dell'idea di base, con qualche spiegazione, e sviluppati in Processing!!!)

<https://github.com/sighack/perlin-noise-fields/tree/master/code> (il codice per ogni esempio)

un tutorial molto semplice e chiaro (per un applicazione di base)

<https://www.youtube.com/watch?v=sZBfLgfsvSk>

<https://www.youtube.com/watch?v=jv6YT9pPIHw> (da capire se utile)

Barney Codes (<https://www.youtube.com/@BarneyCodes>)

<https://p5js.org/reference/p5/noise/> (esempi in p5.js)

(da capire e studiare)

<https://mzucker.github.io/html/perlin-noise-math-faq.html> (obsoleto?)

<https://www.shadertoy.com/view/lsf3WH>

<https://iquilezles.org/articles/> (advanced)

[https://medium.com/@strawberry404/the-fundamentals-of-perlin-noise-bridging-math-and-art-part-1-](https://medium.com/@strawberry404/the-fundamentals-of-perlin-noise-bridging-math-and-art-part-1-8959ee1a55b0)

[8959ee1a55b0](https://medium.com/@strawberry404/the-fundamentals-of-perlin-noise-bridging-math-and-art-part-1-8959ee1a55b0) (vedi anche seconda parte)

<http://devmag.org.za/2009/04/25/perlin-noise/>

<https://jacobstanton.com/generative-typography-a-study/>

<https://www.geeksforgeeks.org/javascript/p5-js-noise-function/>

Machine Learning

TUTTO DA STUDIARE MEGLIO

<https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html>

<https://www.geeksforgeeks.org/machine-learning/machine-learning/>
(da capire quali parti sono interessanti)

https://en.wikipedia.org/wiki/Timeline_of_machine_learning

<https://en.wikipedia.org/wiki/Backpropagation>

<https://en.wikipedia.org/wiki/LeNet> -

- <https://www.geeksforgeeks.org/computer-vision/what-is-lenet/>
- <https://www.geeksforgeeks.org/computer-vision/lenet-5-architecture/>
- <https://www.geeksforgeeks.org/machine-learning/introduction-convolution-neural-network/>

Data

https://en.wikipedia.org/wiki/MNIST_database

<https://quickdraw.withgoogle.com/data>

<https://www.cs.toronto.edu/~kriz/cifar.html>

Personaggi

https://en.wikipedia.org/wiki/Yann_LeCun

<https://awards.acm.org/about/2018-turing>

Corsi universitari (high level)

<https://www.youtube.com/playlist?list=PLoROMvovdv4rMiGQp3WXShtMGgzqpfVfbU>

Stanford CS229: Machine Learning Full Course taught by Andrew Ng | 2018

https://www.youtube.com/playlist?list=PLoROMvovdv4rNyWOpJg_Yh4NSqI4Z4vOYy

Stanford CS229: Machine Learning I Spring 2022Stanford CS229 I

<https://www.youtube.com/watch?v=9vM4p9NN0Ts>

Machine Learning I Building Large Language Models (LLMs) (una lezione)

intro Microsoft

<https://github.com/microsoft/ML-For-Beginners>