CAMILO GOMEZ

PHYSICS WITH EMPHASIS IN DATA SCIENCE

✉ camilogozp@gmail.com
in linkedin.com/in/camilogozp/
github.com/CamGomezDev/
☎ (647) 778 - 9735

SKILLS

Machine Learning

- **TensorFlow**, **PyTorch** and **scikit-learn** used in projects like **materials prediction**, **time series forecasting**, **RL agents** for games, and others
- **Matplotlib** and **Plotly** for graphing
- **Pandas** and **NumPy** to manipulate datasets
- **MLflow** platform for tracking ML experiments, with the **Hyperopt** library for hyperparameter optimization
- **Jupyter** notebooks

Software Engineering

- **Docker** for containerization
- **FastAPI** for microservices
- **Linux** and its terminal
- **Google Cloud Platform** to deploy APIs
- **Git** and **GitHub** for version control
- **unittest** library for testing

Scientific Research

- Abbility to search for relevant **papers** and implement the algorithms in them
- **LaTeX** for writing papers
- **Scientific method** for the creation and analysis of experiments and their results

EDUCATION

Universidad de Antioquia

BSc in Physics

July 2017 - Ongoing

Relevant courses:

- Computational Methods
- Computational Physics I and II (**C++ class** with **Raycaster** as final project)
- Experimental Physics I, II, III and IV
- Neural Networks
- Probability and Statistics

Other

Deep Learning Specialization in Coursera - January 2020

EXPERIENCE

University of Toronto | Toronto, ON

Machine Learning Researcher (May 2022 - August 2022)

- Simulated crystal structures in **PyTorch** with Crystal Graph Convolutional Neural Networks (**CGCNN**) to improve development of new materials.
- Improved and tested neural network models with tools like **hyperparameter optimization**.
- Ran neural networks' pipelines in **cloud clusters** (Compute Canada) with large crystallography datasets.
- Studied and understood crystal structure data from datasets like **Materials Project**.

Guane Enterprises | Medellín, Colombia

Machine Learning Engineer (March 2021 - Present)

- Developed AI application for automatic data extraction of pricesheets documents using **Python** with **Pandas** and ML libraries to accelerate the process done manually by a large American **logistics** company.
- Implemented **NLP** models with **spaCy** for extraction of relevant data in paragraphs.
- Developed all pipelines as **APIs** using **FastAPI**.
- Used **Docker** to deploy everything in containers.
- Communicated with clients mostly located in the **United States**.

Data Analyst (May 2020 - March 2021)

- Created **frontends** that were **component**-based using **Vue.js,** which were connected to APIs for data-focused applications.
- Implemented utilities for ML pipelines like **Levenshtein** similarity algorithm, **outlier detection** for time series, and **forecasts combination** methodologies.
- Used **R** and **RStudio** to do time series forecast optimization.
- Used **Apache ECharts** and **Plotly.js** to create interactive graphs and other ways to display data.

Engineering Intern (February 2020 - May 2020)

- Deployed microservices in the **Google Cloud Platform**.
- Created and improved Python micoservices with **FastAPI**.
- Used **Docker** for the creation of container applications.

PROJECTS

Deep Genetic Snake | github.com/CamGomezDev/DeepGeneticSnake

- Developed a **RL** agent that learned to play the game of Snake using a **neural network** trained with a **genetic algorithm**, built in Java.
- Created the game of Snake from scratch in Java/Processing.
- Agent achieved considerable size before dying.

CAMILO GOMEZ

PHYSICS WITH EMPHASIS IN DATA SCIENCE

camilogozp@gmail.com
linkedin.com/in/camilogozp/
github.com/CamGomezDev/
(647) 778 - 9735

ACCOLADES

- International Physicists' Tournament College Round WInner
- Colombian University Physics Olympiad Third Place
- 2016 Iberoamerican Physics Olympiad Bronze Medal

Smart Tetris | github.com/CamGomezDev/SmartTetris

- Developed agent that learned to play the game of Tetris using a **genetic algorithm**.
- Built the game of Tetris from scratch in the game engine Godot.
- Agent was able to clear 800 lines before dying.

Image Generator with CLIP | github.com/CamGomezDev/NFTs-generator-full-project

- **CLIP neural network** project containerized with **Docker** and **FastAPI**. This architecture is designed to create a picture using only a sentence description.
- Got the FastAPI microservice running in a loop with **Celery** so a set of pictures is created every fixed period of time.
- Added a **front-end** and a **back-end** so user can see the new pictures every new period.

Pokémon Generator | github.com/CamGomezDev/pokemon-generator

- Implemented a **word generator** in **TensorFlow.js** using a **RNN**, which was trained on a Pokemon names dataset.
- A front-end was developed and the application was **deployed** in **Heroku**: poke-generator.herokuapp.com

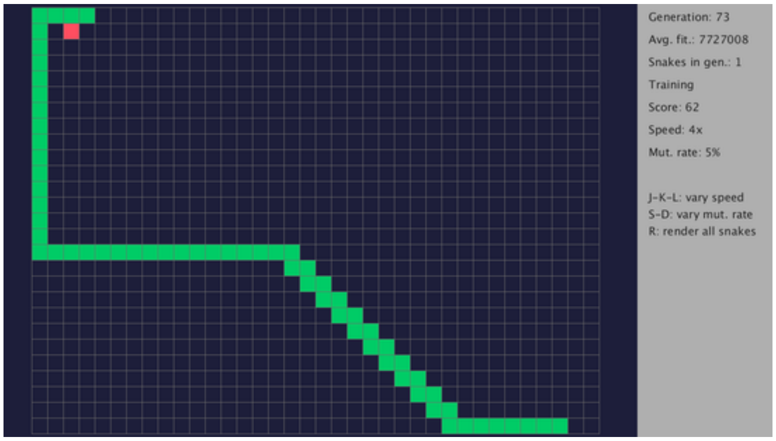
Physics Lab | github.com/CamGomezDev/physics-lab

- Developed a **2D mechanics simulator** in the browser, using **p5.js**.
- Implemented the **physics engine** completely from scratch.
- Deployed using **Heroku**: physicslab.herokuapp.com

YouTube channel | youtube.com/c/CamiloGomezDev

Channel with over 7k subscribers to communicate all of my programming projects online.

DEEP GENETIC SNAKE



```
Generation: 73
Avg. fit.: 7727008
Snakes in gen.: 1
Training
Score: 62
Speed: 4x
Mut. rate: 5%

J-K-L: vary speed
S-D: vary mut. rate
R: render all snakes
```

Figure 1. The snake is the result of training individuals iteratively through generations, where a fitness score is calculated to determine which brains, i.e. neural networks, to crossover in each new generation.

SMART TETRIS

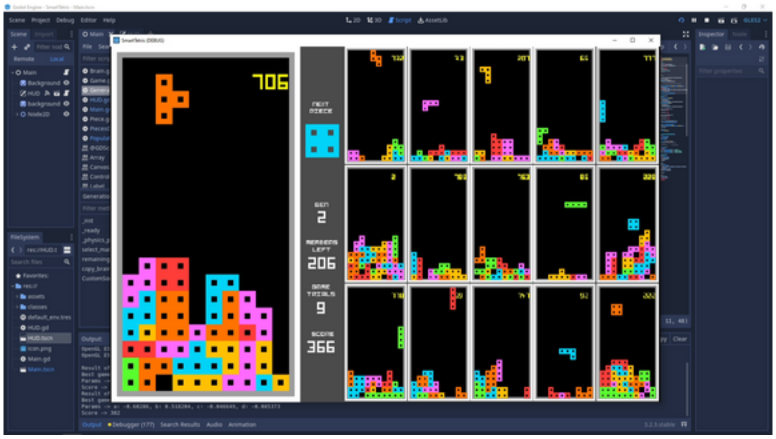


Figure 2. The agent is also trained through a genetic algorithm where the brain is composed of parameters that control the game. All agents of one generation can be seen in the picture.

IMAGE GENERATOR WITH CLIP

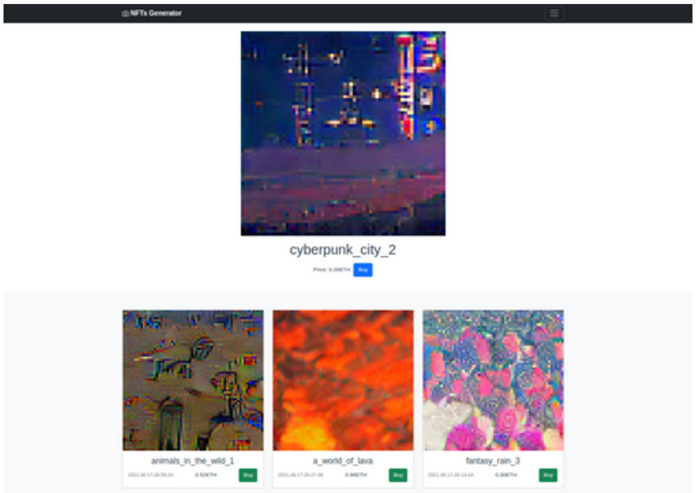


Figure 3. In the picture you can see some of the images generated by the neural network, with a main one being selected for display in the front-end. They're low resolution because a local GPU with little memory was used.

PHYSICS LAB

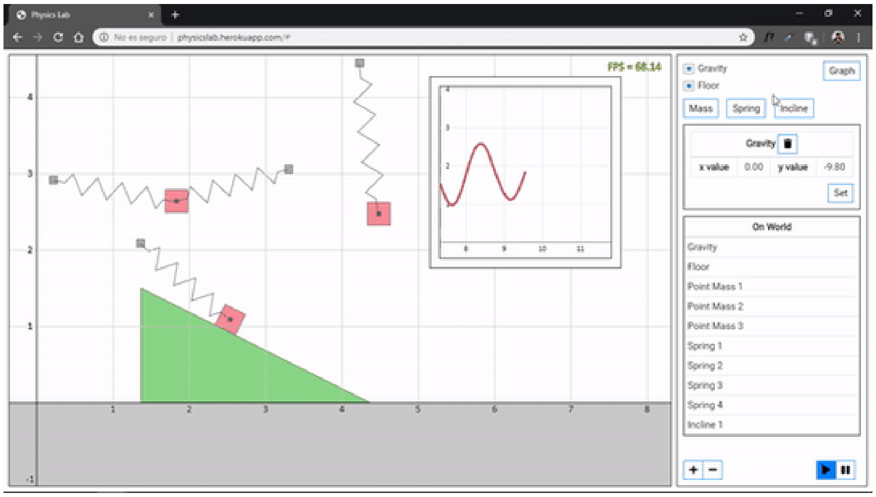


Figure 4. The application is completely interactive. In each new frame, all objects on screen get their forces calculated with the interaction of the objects around them, and then uses an Euler integrator to solve the equations of motion and update the position of each element.