Projet Informatique

IN104

Natalia Díaz Rodríguez, ENSTA Paris, Institut Polytechnique Paris

Logistics

- The most up-to-date material is in the Course repository
- Project preferences: Make 3 choices (in decreasing order of preference) here
  - Deadline to choose one of the 8 topics: 1st April at NOON (1st come, 1st served)
  - Think about your project well, once submitted, your answer cannot be changed.
  - Beware: may not be possible to accommodate everyone; we need balanced groups, respond timely today!
    - Especially with COVID rules, we can’t allocate more students to a salle than their max capacity

Logistics (II)

- IN104 must help you learn manage your autonomy and independence
  - Essential today for an engineer
  - 1 ECTS = 25h of work (all included) -> Expect Min. 2h weekly of extra work at home required
- What are normal grades from past years?
  - No Exam -> Everything between 10 -12 and 20.
  - Up to you to get beyond average
    - A self-learning project
    - Internet is to be used

Logistics (III): Evaluation

1. 25% **Source code**: features, tests, code documentation, etc. Source code must be documented, the report including a link to the repository must be in a .zip file. The repository will contain the sources, as well as a plain text file README.md which indicates the actual operational features and limitations. Python code should compile, the teaching assistants are not supposed to make significant corrections for it to compile: a code with few features, but that compiles and does not crash will be preferred to a more complete code but which is not directly operational!
2. 25% **Defense** (10 minutes presentation, 5 min. questions): The formal quality of the presentation will be an important element. The defense needs to include a demonstration on the basis of the source code, an analysis of the difficulties encountered and implemented solutions. It will not include a presentation of the problem or the method of resolution that the teaching assistant obviously knows already well. The defense is open to everyone (subject to the acceptance of the pair that will present). The chronological order of defense will be given by the list of each group.
3. 20% **Practice Analysis**: You will return a critical (max. 5 pages) report before the defense in which you analyze and criticize the progress of your project and its success and failure factors. This evaluation component also includes the oral treatment of this question during the defense.
4. 30% **Continuous Progress Evaluation** of the practical work during the practical lab sessions (based on Git commits).

Plagiarism control

1. Checks for plagiarism will be performed.
2. Last year, the copying team took a lot of time, conversation, Zoom meetings, and uncomfortable moments, please do not copy code, we find out.
3. Showing signs of copying material is enough reason to have to repeat the course next year.

COVID-19 safety rules

How can we work in pairs if we need to keep 1m distance?

- This is the challenge of COVID era ;)
- You are allowed to communicate via:
  - Voice
  - Sitting in the nearest allowed desk and talking
  - You can communicate the usernames and the repository urls of your project via voice/email/chat.

The same will apply for the rest of the course as work is in binoms (teams of 3 if someone is alone only).

Timetable

- Session 1 Projects presentations and Intro to GIT
    - TD: GIT Practical guide and exercises
- Session 2 Object Oriented Programming (OOP) and Python.
    - TD: Project 0: Unix/Python/ OOP Tutorial: Submit via Gradescope. Deadline: 1 week)
- Session 3 Test Driven Development (TDD).
    - TD: TDD Practical.
- Rest of sessions: Work on Project week
- Last Session Project Defense Day (10 min of soutenance per team including questions)
  Deadline to send report (containing link to project repository) to your Teaching Assistant: Latest 2 days before defense day
- Groups assigned in TD1 are not definitive. They change for the rest of the course in session 2
    - Choose who you want to work with in your project, he needs to have been assigned in your same group, so do not choose until you have been communicated the project you will be working on.

Minimum to pass the course

- Implement and use some **classes** and **objects** in your project
- Implement and run successfully some **unit tests**.
- Play the Git game of push/pull:
  - Coordinate work in the team using **Git** to incrementally progress on the project
  - Your teaching assistant (TA) will check your Github repository *commit, pull, push*… through the course duration

Course expectations

- This is your (first) independent project course
  - i.e., autonomous work expected
  - Time in lab is not expected to be sufficient to hand in a high quality project and report
  - Expected total work hours: at the very least, same amount of lab time, at home
- **Report: 1 single PDF containing link to repository**:
  - Quality over Quantity
  - Results (plots, insights, conclusions and learnings):

more important than lengthy reports explaining the problem

If you feel lost

- Establish milestones with your partner
    - Do GIT code reviews (of each other, if needed)
- Force yourself: pose the right question you want to answer
- Set up Zoom weekly meetings with your binom to work on the project after the TD
- Your TA is there to support you!

If you feel lost

Mid-term report hand in:

- Show to your TA, at least, the first 50% of your project report in the equator of the course
  - makes sure you are on track
  - can help you not derail. You can include:
    - Project Requirements
    - Software Architecture
    - Classes, Tests, Plots, Versions, Documentation...

Course & Team Python Conventions

PEP-8: Defines Python coding practices https://www.python.org/dev/peps/pep-0008

- **Indentation**: 4 spaces (good editors will replace <TAB> by 4 spaces)
- **Variables**: `lower_case_with_underscores`
- **Functions**: `lower_case_with_underscores()`
- **Classes**: `UpperCamelCase`
- **Attributes**: `lower_case_with_underscores`
- **Protected attributes**: `_prefixed_with_1_underscore`
- **Constants**: `ALL_CAPS`
- **Modules**: `lowercase` (single word)

PEP-257: Documentation conventions.

- Prescribes the function or method's effect as a command ("Do this", "Return that:").
- *Docstring* should NOT be a "signature" (reiterating the function/method parameters, which can be obtained by instrospection).

References & Acknowledgements

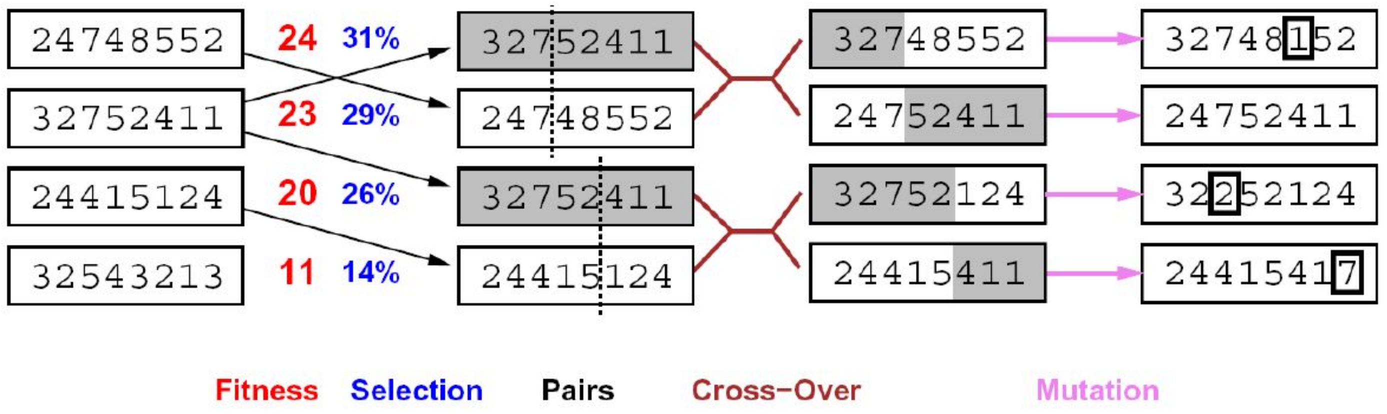
- Jean-Didier Garaud. Coding Best Practices
- Resources to learn Scientific Python: scipy, pandas, numpy: https://github.com/paris-saclay-cds/data-science-workshop-2019
- Lutz, M CC. (2011). Programming Python (4.a ed.). EE. UU.: O’Reilly Media.
- Lutz, M (2013). Learning Python (5.a ed.). EE. UU.: O’Reilly Media.
- Martelli, A (2017). Python in a Nutshell (3.a ed.). EE. UU.: O’Reilly Media.
- Pilgrim, M (2011). Dive Into Python 3(online). APress. http://www.diveintopython3.net
- Python Software Foundation. Python Official Webpage. <http://www.python.org>
- González Duque, R. Python para todos (on line).
- Photography: Sandy Skoglund

5 min Project Pitchs Time!

Project 1: GENETIC ALGORITHMS

Natalia Díaz Rodríguez **Github username:** NataliaDiaz

- Commonly used to generate high-quality solutions to **optimization and search** problems by relying on bio-inspired **operators**
- A **metaheuristic** belonging to the larger class of evolutionary algorithms (EA).
  - Inspired by a natural selection metaphor:
    - Keep best N hypotheses at each step (**selection**) based on a fitness function.
    - Have a pairwise **crossover** operator, with optional **mutation** to give variety

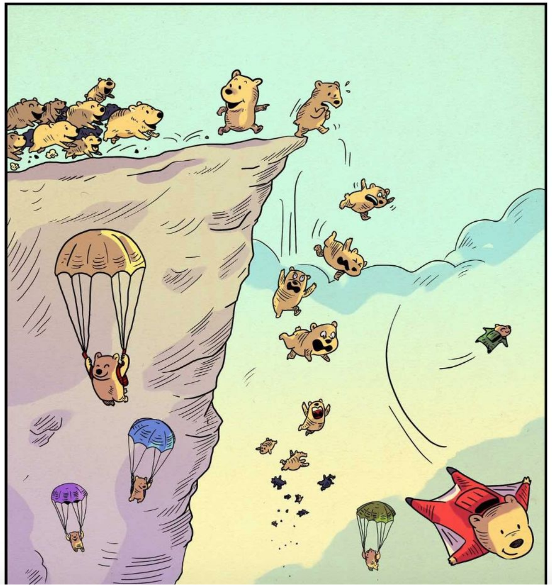


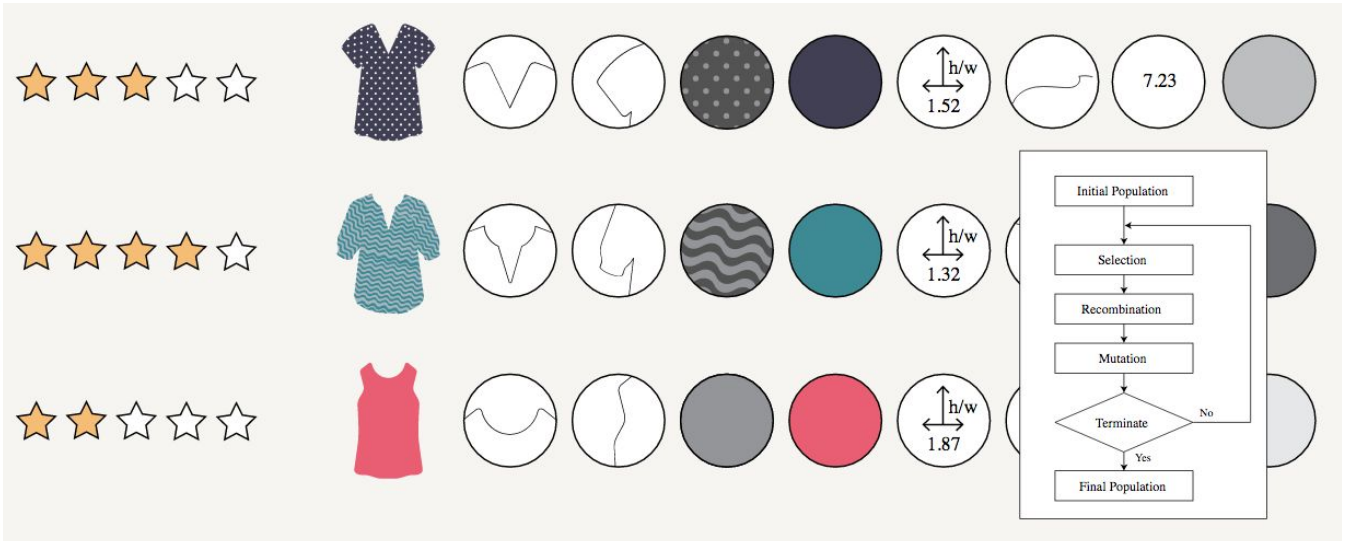
Project 1: GENETIC ALGORITHMS

Natalia Díaz Rodríguez

A Visual Guide to Evolution Strategies

OCTOBER 29, 2017



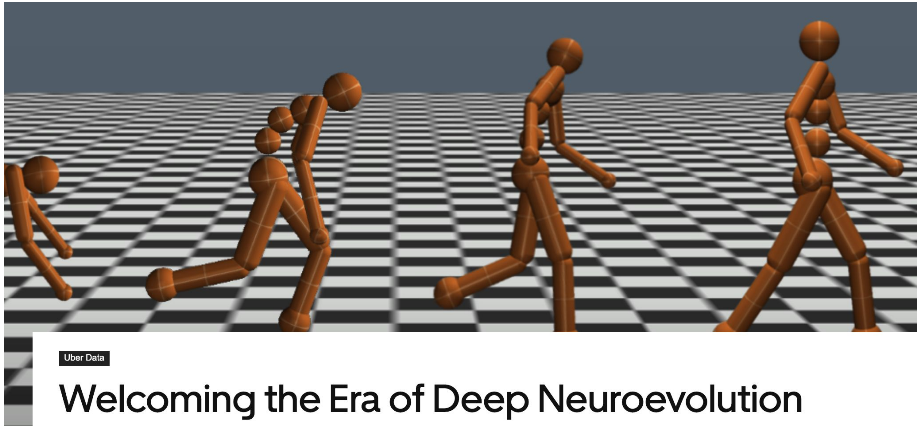


Project 1: GENETIC ALGORITHMS

Natalia Díaz Rodríguez

Intro reads:

- GA: https://en.wikipedia.org/wiki/Genetic_algorithm
- EA: https://en.wikipedia.org/wiki/Evolutionary_algorithm
- Visual Evolution Strategies: http://blog.otoro.net/2017/10/29/visual-evolution-strategies/
- The Era of Deep Neuroevolution (Uber AI Labs): https://eng.uber.com/deep-neuroevolution/
- An intro to genetic algorithms and an example of real industry application on data-driven fashion design https://multithreaded.stitchfix.com/blog/2016/07/14/data-driven-fashion-design/



Contact: natalia.diaz(at)ensta-paris(dot)fr

https://github.com/NataliaDiaz/IN_104-Projet-Informatique

Project 2: SEARCH ENGINE FOR COVID-19

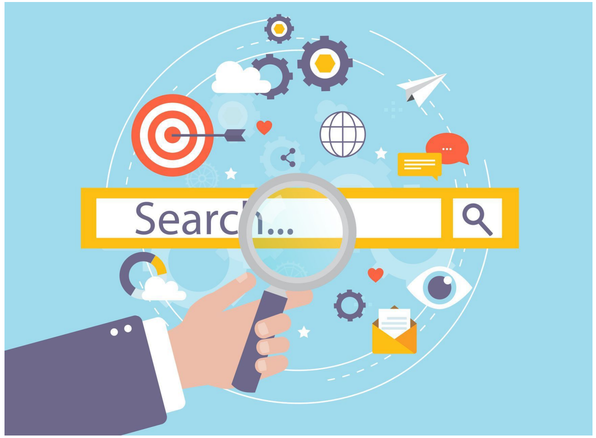
Jessica López Espejel **Github username:** jessicalopezespejel

- A Search Engine for Covid-19 will help the doctors and researchers to find faster articles related to the keywords they are searching for.
- Inspired by the pandemic situation we are living in all the world.

Challenge

- Build your own search engine specialized in Covid-19.
- Use Natural Language Processing strategies.

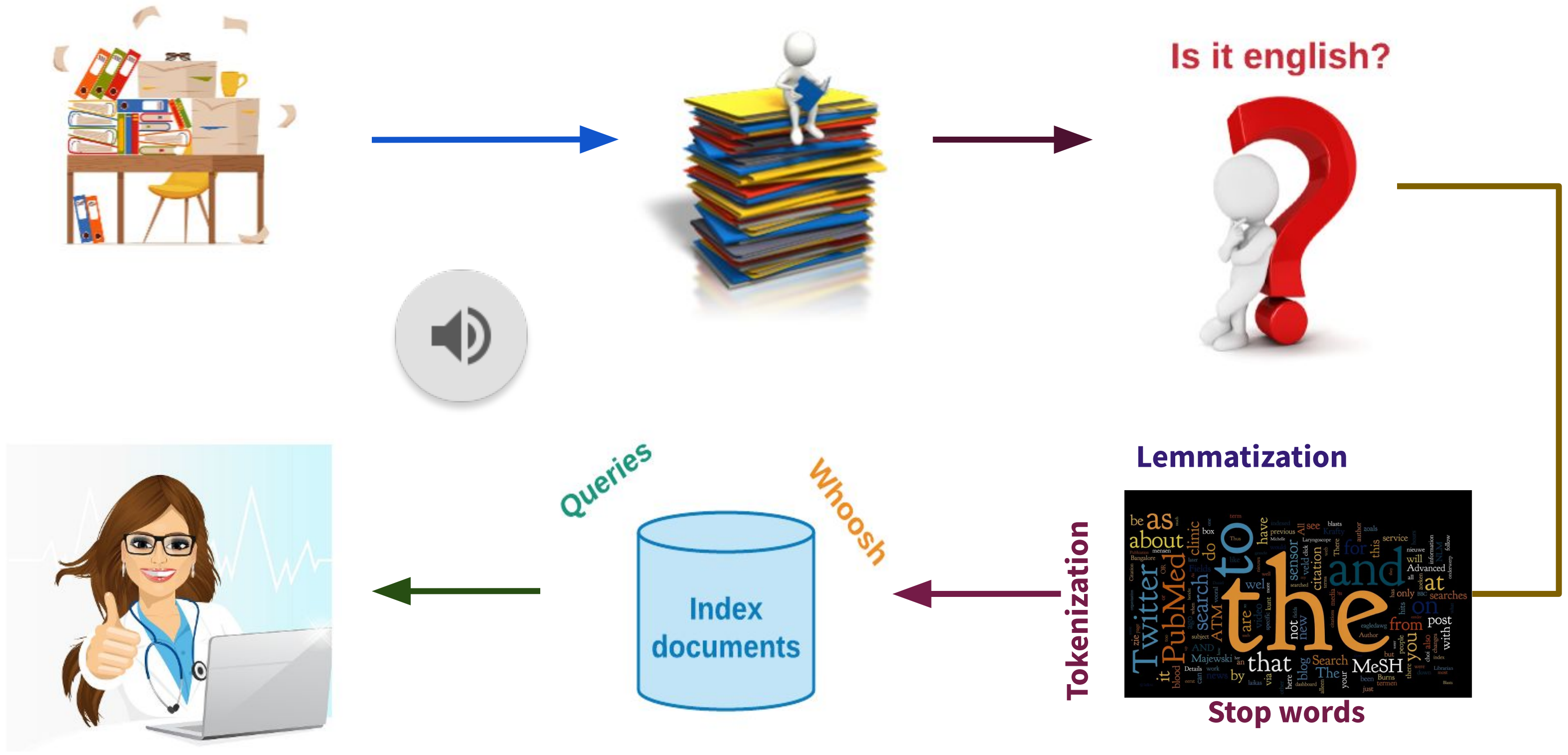
Tools



Python >=3.6; Pandas; Tokenizers such as Scispacy, Spacy, Nltk, etc. Sklearn, Whoosh.

Project 2: SEARCH ENGINE FOR COVID-19

Jessica López Espejel **Github username:** jessicalopezespejel



Project 2: SEARCH ENGINE FOR COVID-19

Jessica López Espejel **Github username:** jessicalopezespejel

Intro reads

- https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html
- https://medium.com/data-science-in-your-pocket/tokenization-algorithms-in-natural-language-processing-nlp-1fceab8454af
- https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8
- https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8
- https://whoosh.readthedocs.io/en/latest/indexing.html
- https://medium.com/@swapnilggourshete/how-indexes-on-database-works-15efa266fda6
- https://medium.com/@abhilashbss/how-indexing-works-in-databases-931c8a73ea42

Contact: jessicalopezespejel(at)gmail(dot)com

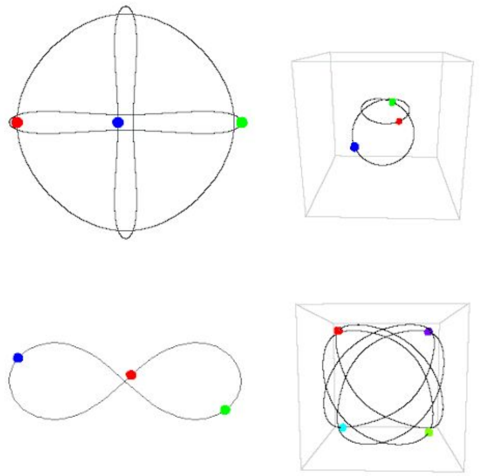
More details: **https://github.com/JessicaLopezEspejel/N_104-Projet-Informatique**

Ismail Bennani <ismail.lahkim.bennani@ens.fr>

Project 3: N-Body problem

Simulate a (very) tiny universe

- "The n-body problem is the problem of predicting the individual motions of a group of celestial objects interacting with each other gravitationally" - Wikipedia



You will learn about:

- Physics engines
- Numerical integration schemes:
  - approximate solutions of Ordinary Differential Equations
  - inherent tradeoff between speed and accuracy
- 2D/3D drawing:
  - basic 2D drawing primitives
  - 3D projection on a 2D plane

GIFs from http://rebloggy.com/post/gif-gifset-science-math-physics-n-body-problem-orbits/39788206329

Examples: https://vanderbei.princeton.edu/WebGL/nBody.html

Ismail Bennani <ismail.lahkim.bennani@ens.fr>

Project 3: N-Body problem

Simulate a (very) tiny universe

Your job (mandatory):

- Write a physics engine computing the forces applied to each body
- Write a an ODE solver to integrate the dynamics
- Write a 2D graphical interface to show the results

Tool suggestions	
Language	Python
2D drawing lib	Pygame

The rest is up to you, you can:

- Compute physics faster, e.g. simulate an asteroid belt (700.000 to 1.7M asteroids in our solar system)
- Compute more accurate dynamics (Leapfrog algorithm, Runge-Kutta family, …)
- Handle collisions between bodies
- Add features to the GUI, e.g. control the camera with your mouse, draw trajectories of bodies, …
- Simulate the systems in a 3D space and project it on screen
- … be creative !

More details: https://github.com/ismailbennani/IN104/blob/master/sujet/sujet.pdf

Git repo: https://github.com/ismailbennani/IN104

Github username: ismailbennani

Project 4: Titanic survivor prediction and explainable artificial intelligence techniques (Thomas Rojat thomas.rojat@renault.com)

•Projet de machine learning

•En utilisant la base de donnés du titanic, prédire quels seront les passagers qui réussiront à survivre à l'aide d'un modèle de machine learning

•Le but du projet est d'expliquer, de donner des informations sur ce que le modèle a appris (sur quoi se base-t-il pour prendre ses décisions?)

https://docs.google.com/document/d/1GwMIzYUrvZmVonxz19mEBA-CKCUsRCQ2kmG4gqppHnw/edit?usp=sharing

Project 4: Titanic survivor prediction and explainable artificial intelligence techniques (Thomas Rojat thomas.rojat@renault.com)

- Le choix du modèle est laissé libre (Random Forest, SVM)
- Des exemples d’implémentation avec la bibliothèque sklearn seront fournis
- Pour l’explicabilité, deux méthodes devront être utilisées(des exemples de code seront fournis) : Partial Dependence Plot (PDP) et accumulated local effects (ALE)
- Explorer ce que peut apporter PDP et ALE en terme d’explicabilité

Project 5: Trading Bot in the European Weather and Energy Markets

Eliot Tabet

- Compréhension du marché de l’énergie (gaz/électricité) et l’impact du climat dans ce marché
- Prévision de la consommation
- Prise de décision sur la position à prendre (Buy/Sell) en se basant sur les algorithmes de machine learning

https://drive.google.com/drive/folders/1o9m_fSyRIx9L26HXHe1cnxhT_F9vnxzB?usp=sharing

eliot.tabet@polytechnique.edu

Project 5: Trading Bot in the European Weather and Energy Markets

Eliot Tabet

- Python: Pandas, Numpy, SciPy, SkLearn, Matplotlib, dfply
- Supervised Learning Algorithms (Classification and Regression): Random Forest, Logistic Regression, Non-linear regression, Multidimensional Linear Regression

https://drive.google.com/drive/folders/1o9m_fSyRIx9L26HXHe1cnxhT_F9vnxzB?usp=sharing

eliot.tabet@polytechnique.edu

Project 6: Build a navigation app for your phone*

Stephen Creff

*the smartphone is not provided

- Use native or cross-platform app development framework to build an iOS or Android app,
- Choose your language (Python, Java, …) and find your development framework (tech stack),
- Understand the framework and the APIs (widgets, …)

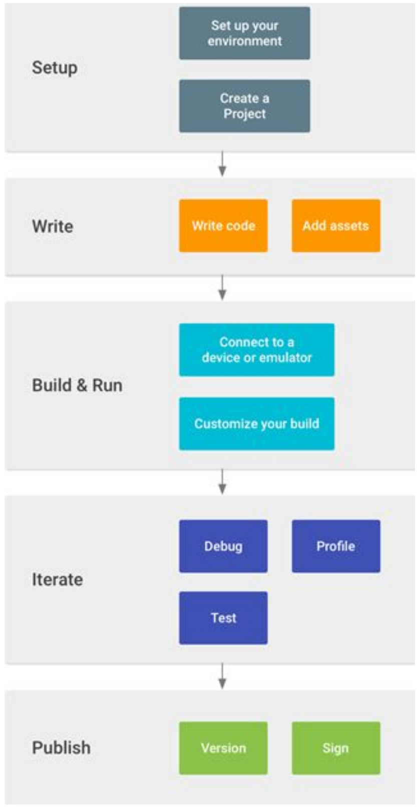
- Make your System Analysis and Requirements,make it incremental!
  - Add features and sensors solicitation step by step, from trivial to rich location-based
  - Define map (simplified, real, feature-rich), set geo-markers, calculate and draw route, …
- Define a storyboard and mock-up the Interface design
- Define and build functions using smartphone sensors
  - Touch screen, Geolocation service, …
- Package & deploy your app on the targeted OS

stephen.creff@gmail.com
https://github.com/ctfun/IN104_2020

Project 6: Build a navigation app for your phone*

Stephen Creff

*the smartphone is not provided



Key points of the project:

- Follow a full system development workflow,
- Learn to program and use dedicated APIs,
- Routing problem, pathfinding algorithms,
- *And, have fun with your phone !*

Some frameworks to build Mobile Applications:

- A python framework : https://kivy.org/#home
- A java framework : https://www.codenameone.com/
- Other framework, with combination of many languages i.e., HTML5, JavaScript, and CSS and Cordova wrapper : https://ionicframework.com/

Project 7: Artificial Intelligence for 2 players games

Clement Masson / Computer Vision and Deep Learning engineer at Thales

You will learn the basics of algorithms used for most 2 players board games.

(disclaimer: this is NOT a Machine Learning project. It would require too much time to tackle the ML part)

masson.cle@gmail.com

Project 7: Artificial Intelligence for 2 players games

Clement Masson / Computer Vision and Deep Learning engineer at Thales

You will learn the <u>basics</u> of algorithms used for most 2 players board games.

(disclaimer: this is <u>NOT</u> a Machine Learning project. It would require too much time to tackle the ML part)

masson.cle@gmail.com

Project 7: Artificial Intelligence for 2 players games
Clement Masson

Language: python

Expected work: project will start with many guidelines so you know where to start.

Evaluation:

- Oral presentation & report : 50 %
- AI Competition ranking : 40%
- Code quality, good use of git, deadline respect, work in group … : 10 %

Useful links:
Project website (forum, guidelines, help, …): https://sites.google.com/view/ensta-in104
Wiki : https://github.com/clement-masson/aiarena/wiki

masson.cle@gmail.com

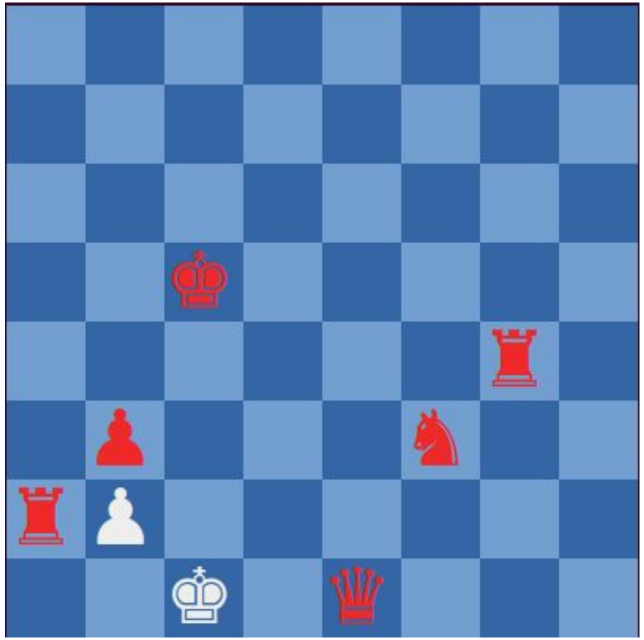
Project 7: Artificial Intelligence for 2 players games

Fun :

- You can play against your own AI
- Your AI will compete against other groups' AI !

Interesting theory aspects :

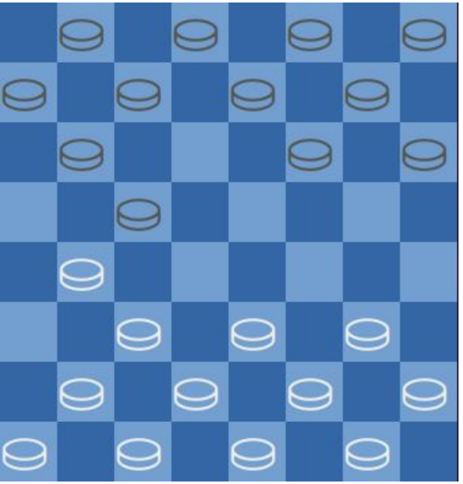
- Recursivity, Trees
- Minimax, alpha-beta pruning
- Object Programming
- Code profiling and optimization

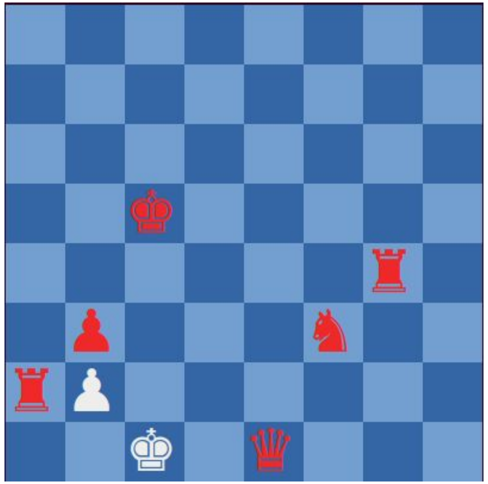


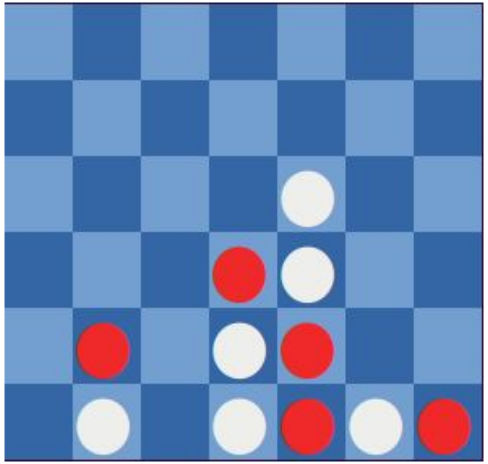
masson.cle@gmail.com

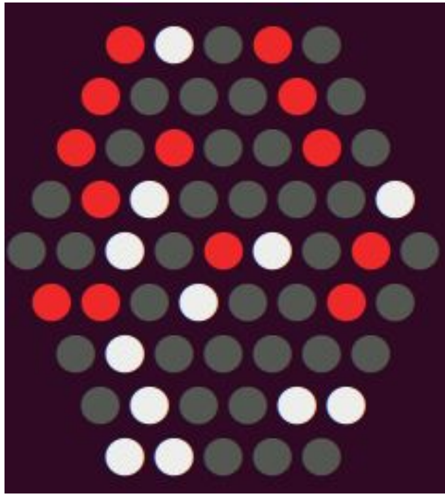
Project 7: Artificial Intelligence for 2 players games
Clement Masson

Some code is already provided to let you quickly start your implementation of AIs and play against them !


checkers


chess


connect4


abalone

masson.cle@gmail.com

Recap IN104

Link to slides and course material:
https://github.com/NataliaDiaz/IN_104-Projet-Informatique

Lecture 1: Intro to GIT: Work in pairs for Session 1

Lecture 2: Intro to Object Oriented Programming (OOP): Groups change here for the rest of the course, choose who you want to work with in your project.

Lecture 3: Test Driven Development (TDD) in Python

Practical work (TD): indicated at the end of each Lecture slides