

# Open Project : Deep Learning et Voiture Autonome

---

## Abstract:

---

Le projet de voiture autonome de 42AI est devenu virtuel a cause du coronavirus. Nous allons creer et entrainer un agent de Deep Learning afin de pouvoir **conduire la voiture virtuelle** en se basant sur l'image d'une camera apposee sur le toit de la voiture.

Nous avons elu d'approcher le probleme par la voie du **Reinforcement Learning** car la virtualisation de l'environnement nous le permet. La course se passe dans un environnement nomme DonkeyCar.

## Parts of the project:

---

### 1. Interface DonkeyCar

Nous devons ecrire un programme qui nous permet de controller une voiture dans l'environnement DonkeyCar et de recolter les informations generees par le-dit environnement (i.e. Vitesse de la voiture, distance parcourue etc).

### 2. Recherche de sponsors

Le Reinforcement Learning nécessite beaucoup de puissance de calcul. Si nous pouvons trouver un sponsor pour le projet, ce qui n'est pas garanti, cela nous garantira l'accès à une puissance de calcul suffisante.

### 3. Mise en place de metriques de qualite

Afin de pouvoir comparer différentes solutions possibles, de pouvoir quantifier l'impact de certaines modifications et de pouvoir suivre l'évolution du projet il nous faut definir des metriques capables d'évaluer un "conducteur".

Tres clairement il nous faut des fonctions: *metrique(course) => score*

### 4. Etat de l'art et choix d'un modele

Il nous faut dans un premier temps un état de l'art des solutions de Deep Learning (plus spécifiquement de RL) afin de pouvoir considerer les différentes solutions possibles. Ceci implique une lecture et une archivation de la littérature scientifique actuelle. Ensuite nous allons choisir une ou plusieurs solutions et les appliquer à notre problématique.

### 5. Data Engineering

Des formats de données, solutions de stockages et des interfaces doivent être établis pour les différentes parties du projet afin de permettre une collaboration fluide. Potentiellement une solution de monitoring sera mise en place

### 6. Creation du modele

Le modele doit être cree et entraîné.

### 7. Optimisation / Creation d'un ensemble

Le modele choisi va être entraîne et optimise (par ses hyperparametre et sa topologie) afin d'en optimiser la performance. Un ensemble peut aussi être considere (i.e. plusieurs modeles a la suite. Par exemple un modele pour le traitement de l'image puis un modele "conducteur" puis un modele "évaluateur" travaillant en tandem)

### 8. Course

DonkeyCar US organise des courses dans son environnement mensuel. Il faudra mettre en place un serveur capable de faire tourner notre modele et d'interfacer avec leur serveur.

### 7. Plus

1. Faire apprendre au modele a gerer une latence.
2. Faire de l'augmentation de dataset pour être plus efficace en learning par calcul.
3. Stocker nos données sur un serveur afin de les rendre accessibles à tous les membres du projet.
4. Optimisation automatique des hyperparametres
5. Paralelisation de l'apprentissage

# Stack Technique

---

## Interface DonkeyCar:

- Donke Car Simulator: <https://docs.donkeycar.com/guide/simulator/>
- gym environment: <https://github.com/tawnkramer/gym-donkeycar>

## Preprocessing en python:

- numpy
- tensorflow
- keras

## Modele en python:

- numpy
- tensorflow
- keras

## Data Engineering en python:

- numpy
- keras, tensorflow
- pickle
- AWS S3 / IAM

## Metriques d'evaluation et optimisation en python:

- Numpy

## Course:

- Donkey Car Simulator: <https://docs.donkeycar.com/guide/simulator/>
- gym environment: <https://github.com/tawnkramer/gym-donkeycar>