



FORUM INNOVATION
INGÉNIERIE | INFORMATIQUE |
ENTREPRENEURIAT | **UQAR**

Simulation 3D de stationnement automatique **par Brandon Gauthier**

Présentés au FI3E par le Cégep de Matane

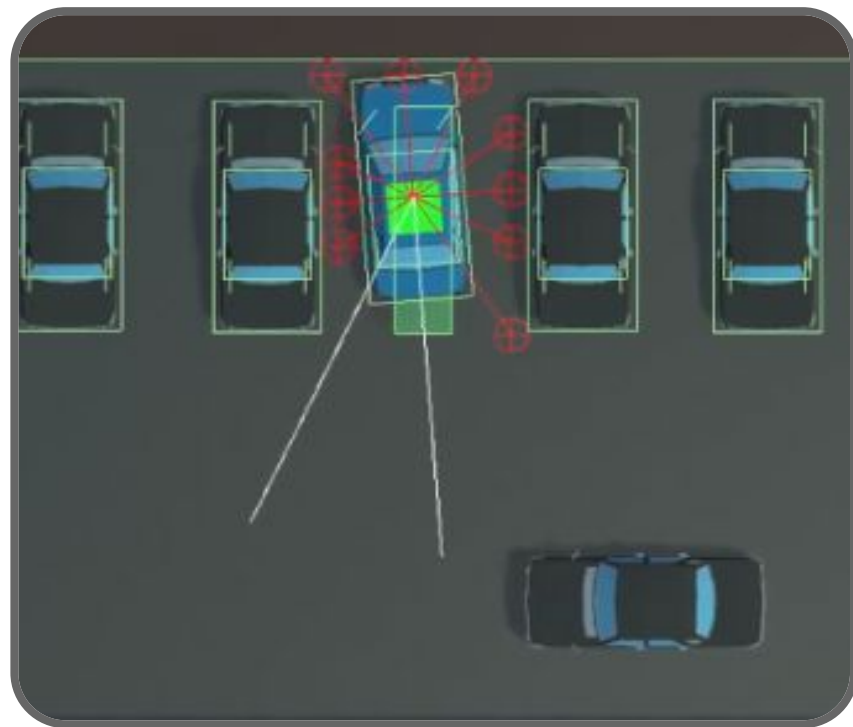


FORUM INNOVATION
INGÉNIERIE | INFORMATIQUE |
ENTREPRENEURIAT | UQAR

Simulation 3D de stationnement automatique

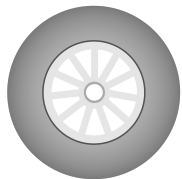
par **Brandon Gauthier**

Une voiture apprend à se stationner par elle-même grâce à un algorithme de neural network et la PPO dans une simulation tridimensionnelle réalisée sur Unity.

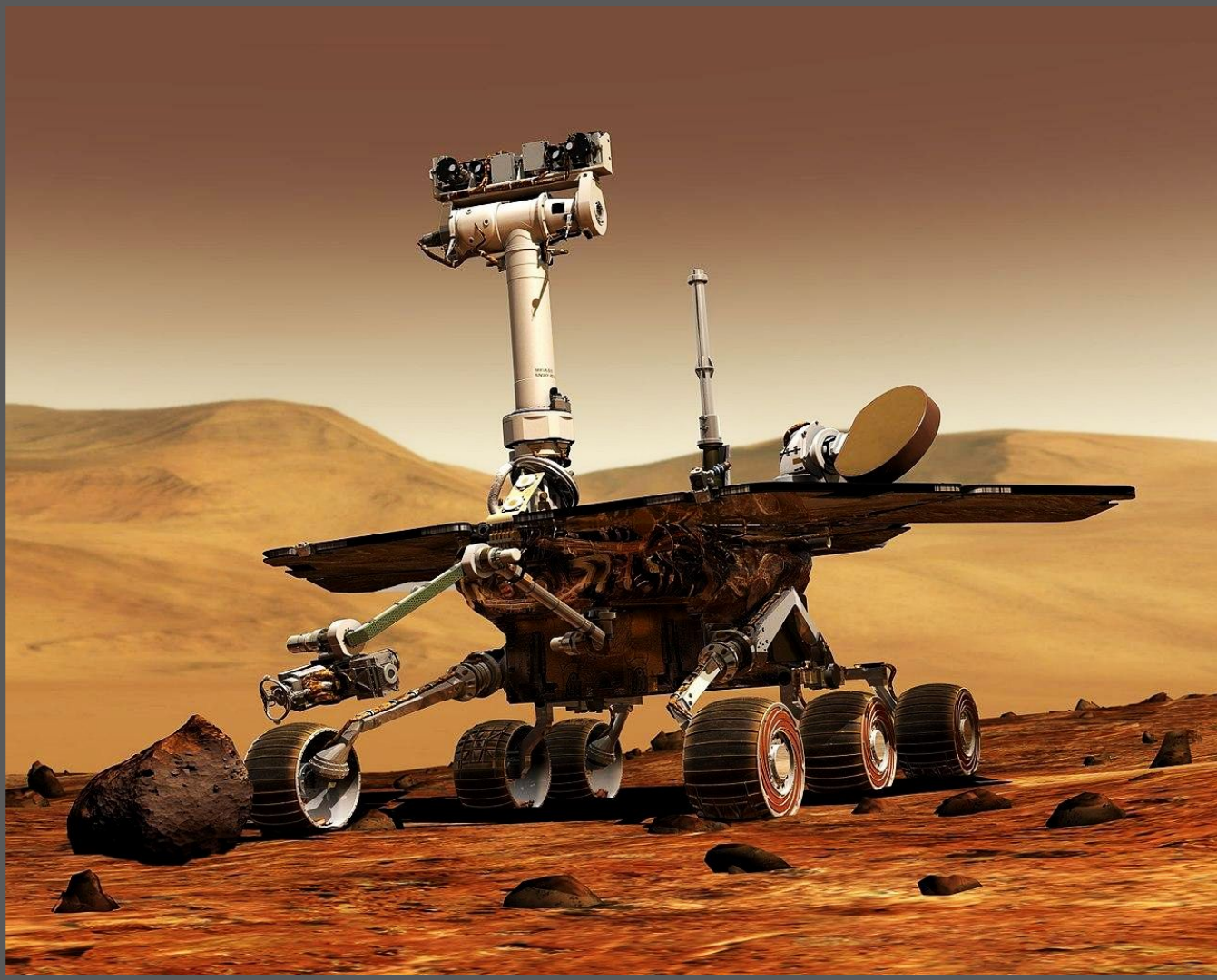


Pourquoi ?

La navigation autonome dans des environnements complexes pose des défis majeurs, tant pour les véhicules terrestres que pour l'exploration spatiale.



Tester et perfectionner ces technologies dans le monde réel est coûteux et risqué.



Pourquoi

Ce projet utilise des simulations avancées pour développer et affiner les algorithmes de prise de décision.

Cette méthode permet de préparer efficacement les systèmes de navigation autonome pour des applications réelles, réduisant ainsi les risques et les coûts associés aux tests physiques.

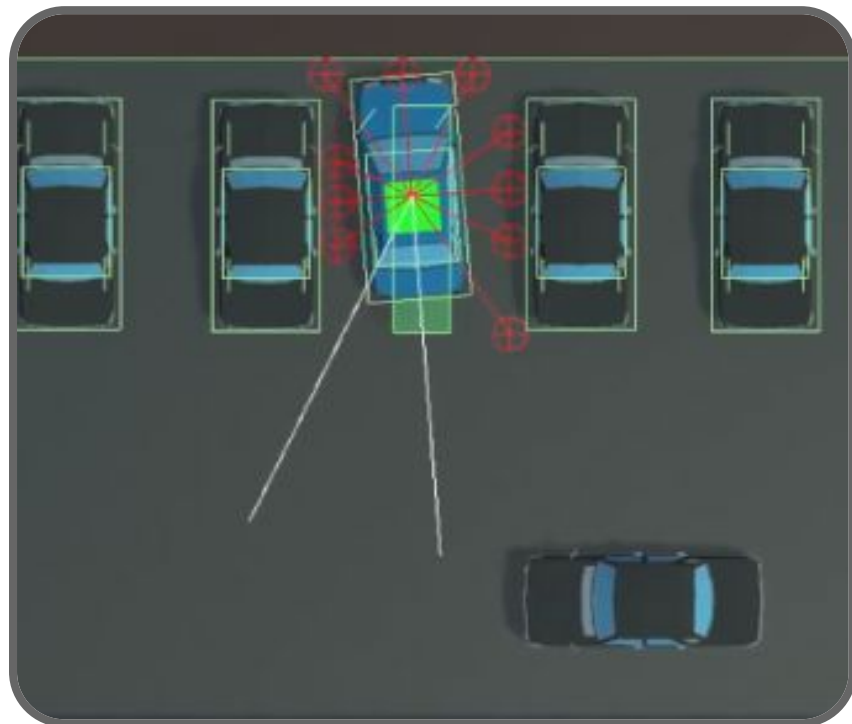


Illustration par l'intelligence artificielle

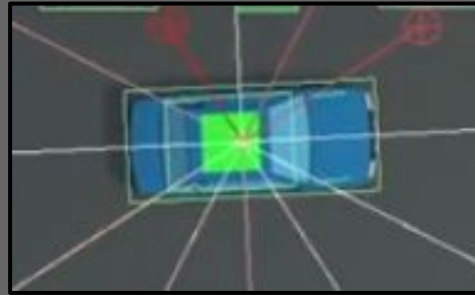
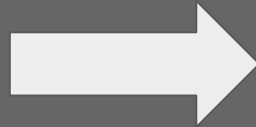
Objectifs

Le but de ce projet est de développer un agent IA capable de se stationner de manière autonome dans un environnement de parking simulé.

Un objectif technique est de démontrer l'efficacité de l'algorithme PPO dans la prise de décision de navigation. Enfin, les hyperparamètres doivent être optimisés afin d'éviter le surapprentissage (overfitting) et le sous-apprentissage (underfitting).



MÉTHODOLOGIE

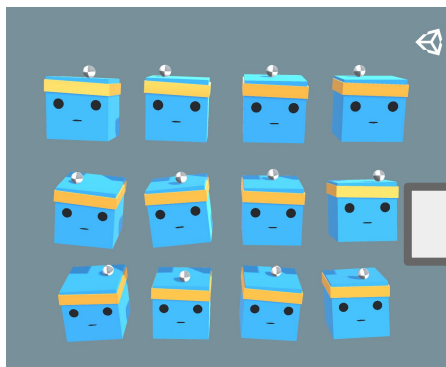


Méthodologie

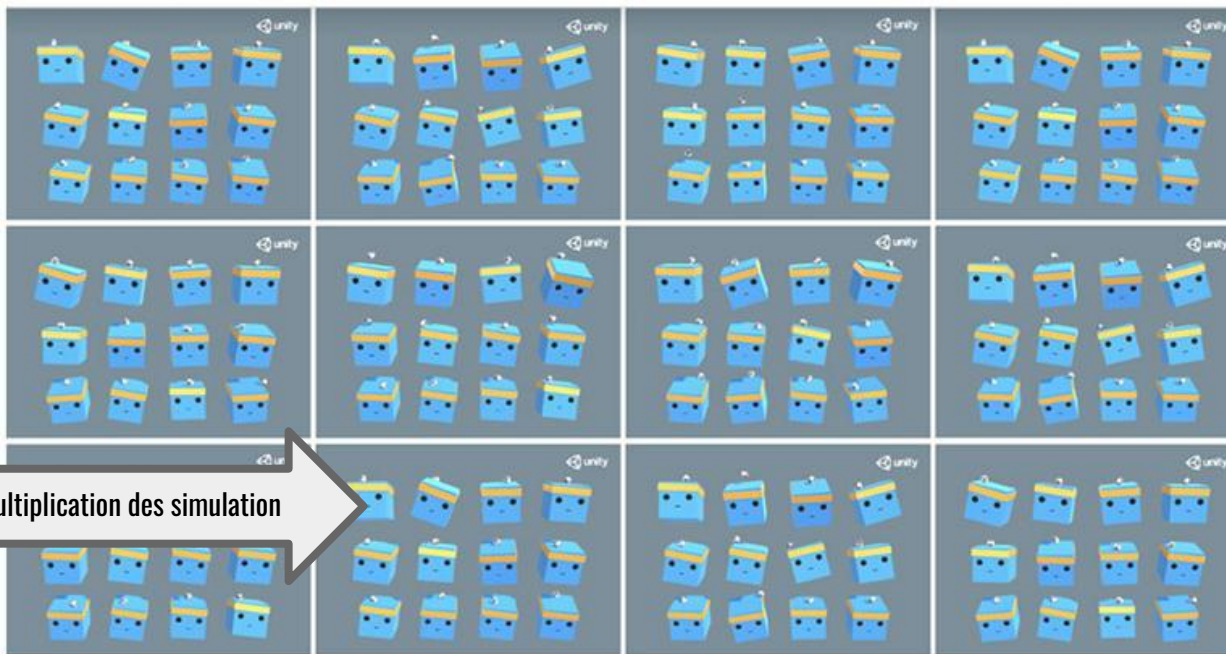
- Création d'un environnement virtuel interactif pour entraîner l'intelligence artificielle
- Élaboration d'une démo pour que l'intelligence artificielle puisse avoir un point de référence
- Sélection de paramètres de départ : 3 layers, 264 neurones, learning_rate de 0.00035, reward gain de 0.3
- Entraînement de l'intelligence de l'agent.
- Ajustement des hyperparamètres en fonction des résultats - Répéter les 2 dernières étapes.

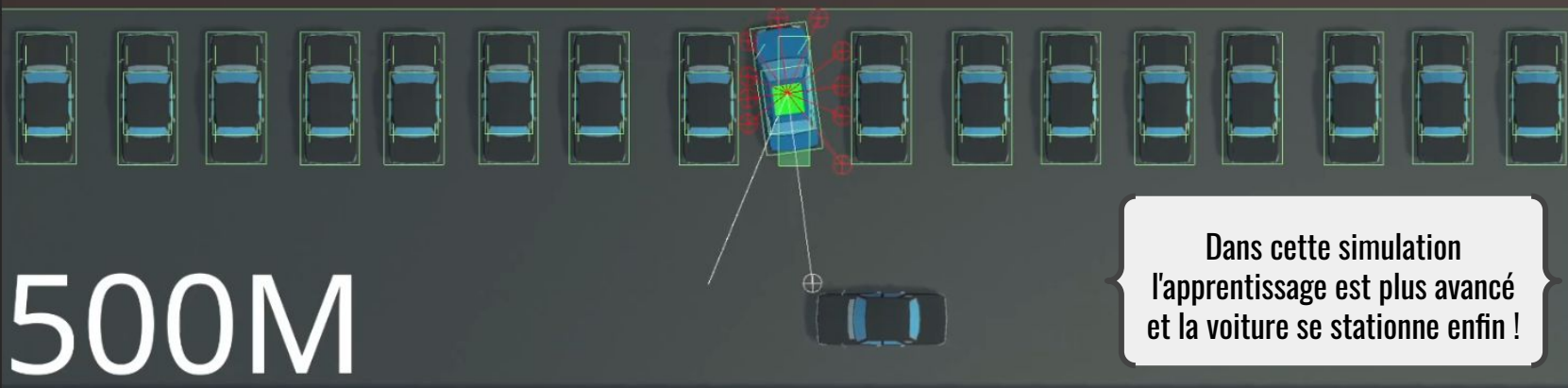
Méthodologie

Avec ML-Agent une seule simulation peut être roulée en parallèle plusieurs fois.



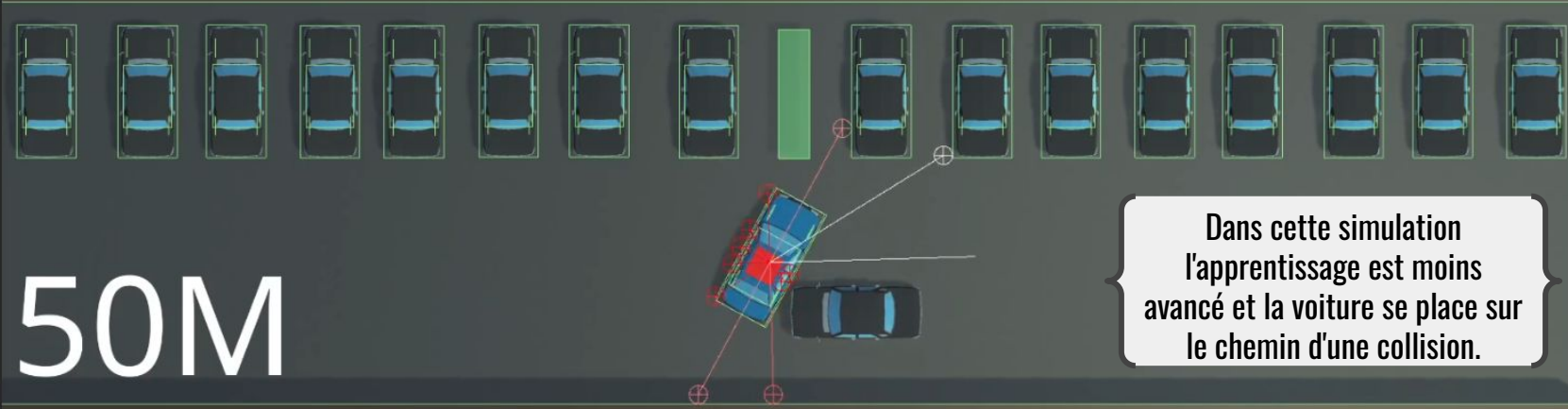
Multiplication des simulation





500M

Dans cette simulation l'apprentissage est plus avancé et la voiture se stationne enfin !



50M

Dans cette simulation l'apprentissage est moins avancé et la voiture se place sur le chemin d'une collision.

TensorBoard

L'utilisation de TensorFlow et TensorBoard est essentielle pour affiner les paramètres d'un modèle d'apprentissage par renforcement qui est entraîné à effectuer des tâches complexes telles que stationner une voiture dans un environnement de simulation Unity.

En observant les graphiques générés par TensorBoard, on peut ajuster les paramètres de l'IA pour améliorer ses performances.

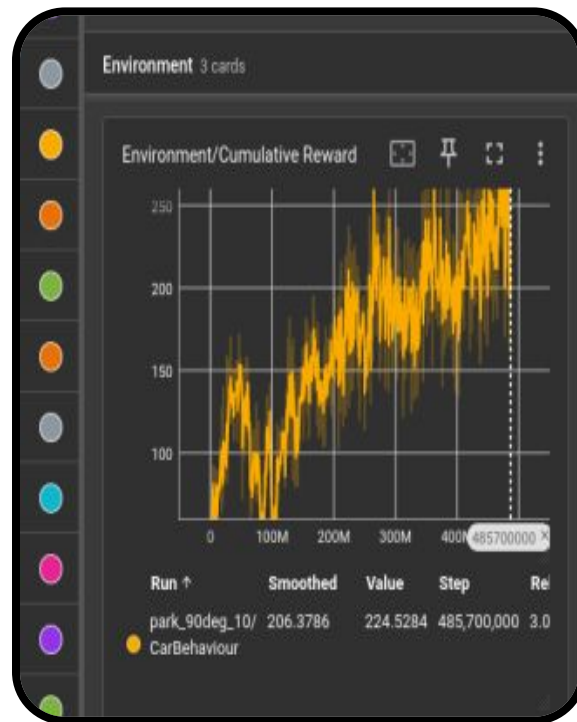


Le graphique de récompense (reward)

Le graphique de récompense cumulative montre comment l'IA apprend au fil du temps.

Si les valeurs augmentent de façon stable, cela suggère que l'IA est en train de s'améliorer.

Une stagnation ou une baisse indiquerait qu'il faut peut-être ajuster le taux d'apprentissage ou explorer des politiques d'exploration plus efficaces.

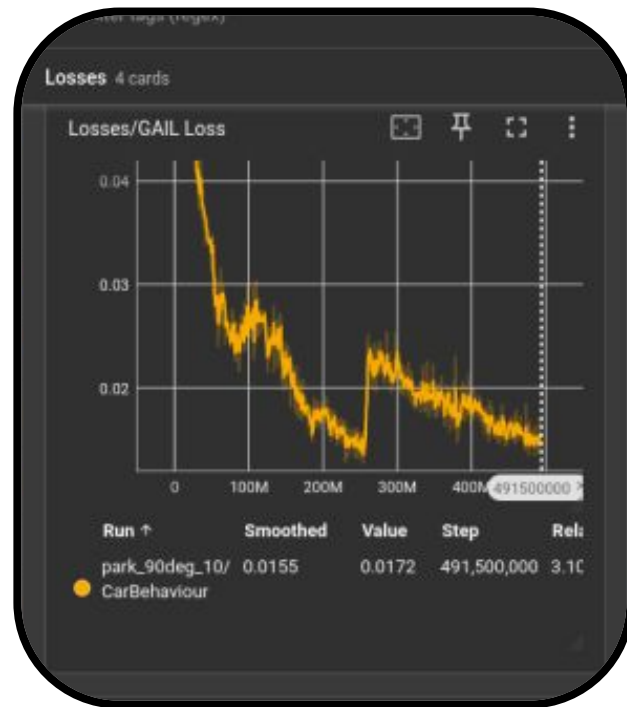


Le graphique de pertes (loss)

Les graphiques de perte, tels que GAIL Loss, Policy Loss et Value Loss, sont des indicateurs de la façon dont l'IA optimise son apprentissage.

Une perte qui diminue indique un apprentissage efficace.

Si une perte cesse de décroître ou augmente, cela peut signifier que l'IA a atteint un minimum local ou que les hyperparamètres nécessitent des ajustements, comme le taux d'apprentissage ou les poids de régularisation.



Configuration de départ

behaviors:

CarBehaviour:

trainer_type: ppo

hyperparameters:

batch_size: 1024

buffer_size: 5120

learning_rate: 0.00035

beta: 0.0025

epsilon: 0.3

lambda: 0.95

num_epoch: 5

learning_rate_schedule: linear

network_settings:

normalize: true

hidden_units: 264

num_layers: 3

reward_signals:

extrinsic:

gamma: 0.95

strength: 0.99

gail:

strength: 0.3

demo_path: ./Demos/90dgRS3s.demo

use_actions: false

keep_checkpoints: 15

checkpoint_interval: 1_000_000

time_horizon: 264

max_steps: 500_000_000

summary_freq: 100_000

threaded: true

env_settings:

env_path: ./build_90deg_50x_1/build_90deg_50x_1.x86_64

env_args: null

base_port: 5005

num_envs: 5

num_areas: 1

engine_settings:

no_graphics: true

checkpoint_settings:

run_id: park_90deg_10

force: false

results_dir: results

torch_settings:

device: cuda

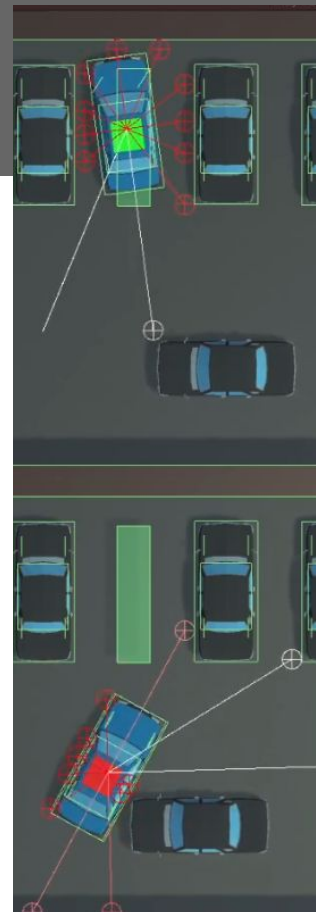
RÉSULTATS

Résultats

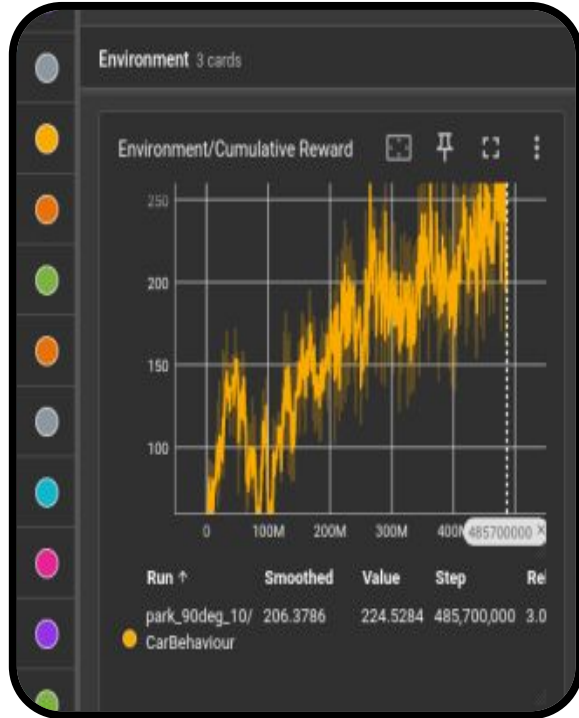
L'intelligence artificielle est capable de se stationner automatiquement dans certaines positions de départ.

Dans ces situations, le projet a permis de démontrer l'efficacité du modèle PPO dans un scénario de navigation complexe, où l'agent a réussi à apprendre et s'adapter à des conditions dynamiques & imprévisibles.

Les graphiques de TensorBoard ont illustré une progression généralement constante dans l'apprentissage de l'agent, validant ainsi l'approche utilisée pour l'entraînement. En superposant les vidéos de 50 essais versus 500 essais, on voit à l'oeil nu que l'intelligence la plus entraînée est celle qui réussit à se stationner contrairement à l'autre.



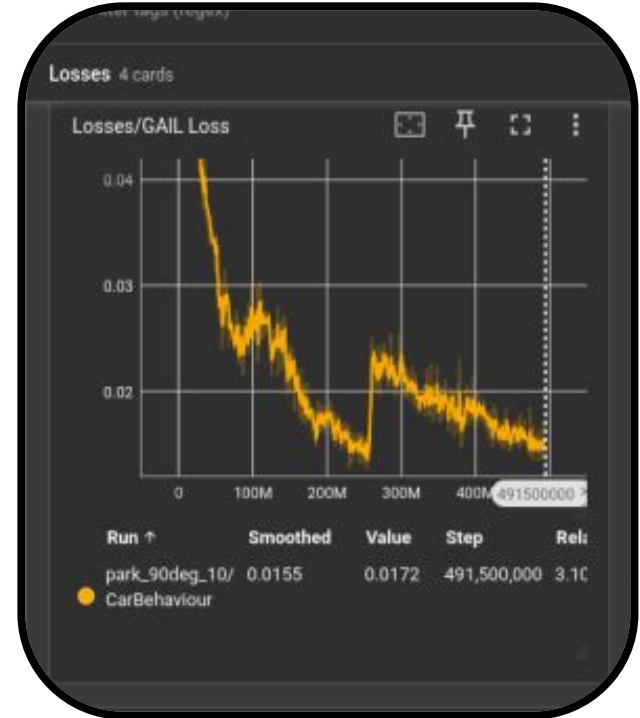
Résultats - Graphiques de TensorFlow



Dans le graphique de récompense cumulative montre que les valeurs augmentent de façon stable, cela suggère que l'IA apprend bien au fil du temps.

Dans le graphique de PERTES Gail Loss, on voit que les pertes (l'erreur) diminue au fil du temps. Ceci indique un apprentissage.

Cependant on voit une augmentation soudaine, ce qui indique qu'il est ressortit d'un minimum local et a reconvergé.



Références

Vanlsegghem Thomas - AI-Parking-Unity

<https://github.com/VanlsegghemThomas/AI-Parking-Unity>

Vanlsegghem Thomas - Automated Parking Using RL, a Unity ML-Agents Tutorial

https://www.youtube.com/watch?v=_Bzw2B-9QkM

Joy Zang - Proximal Policy Optimization (PPO) with Unity ML-Agents

<https://www.gocoder.one/blog/training-agents-using-ppo-with-unity-ml-agents/>

Thomas Simonini - Proximal Policy Optimization (PPO)

<https://huggingface.co/blog/deep-rl-ppo>

Unity ML-Agent Toolkit

<https://unity-technologies.github.io/ml-agents/Getting-Started/>

<https://unity.com/products/machine-learning-agents>