

# **DONKEY BOMBS**

# **STRATÉGIE DE CONCEPTION** **DE L'AGENT**

**OBJECTIF :** Concevoir un agent stable et efficace

**APPROCHE :** Du fonctionnel à l'optimal

**PRINCIPE DE  
DIRECTION :**  
SECURITY > PERFORMANCE

# MID PILOT

## La méthode `position_track`

- Entrée : `paths_end, velocity`
- Sortie : On retourne l'écart latéral x et l'écart avant z du point cible

## La méthode `compute_turning`

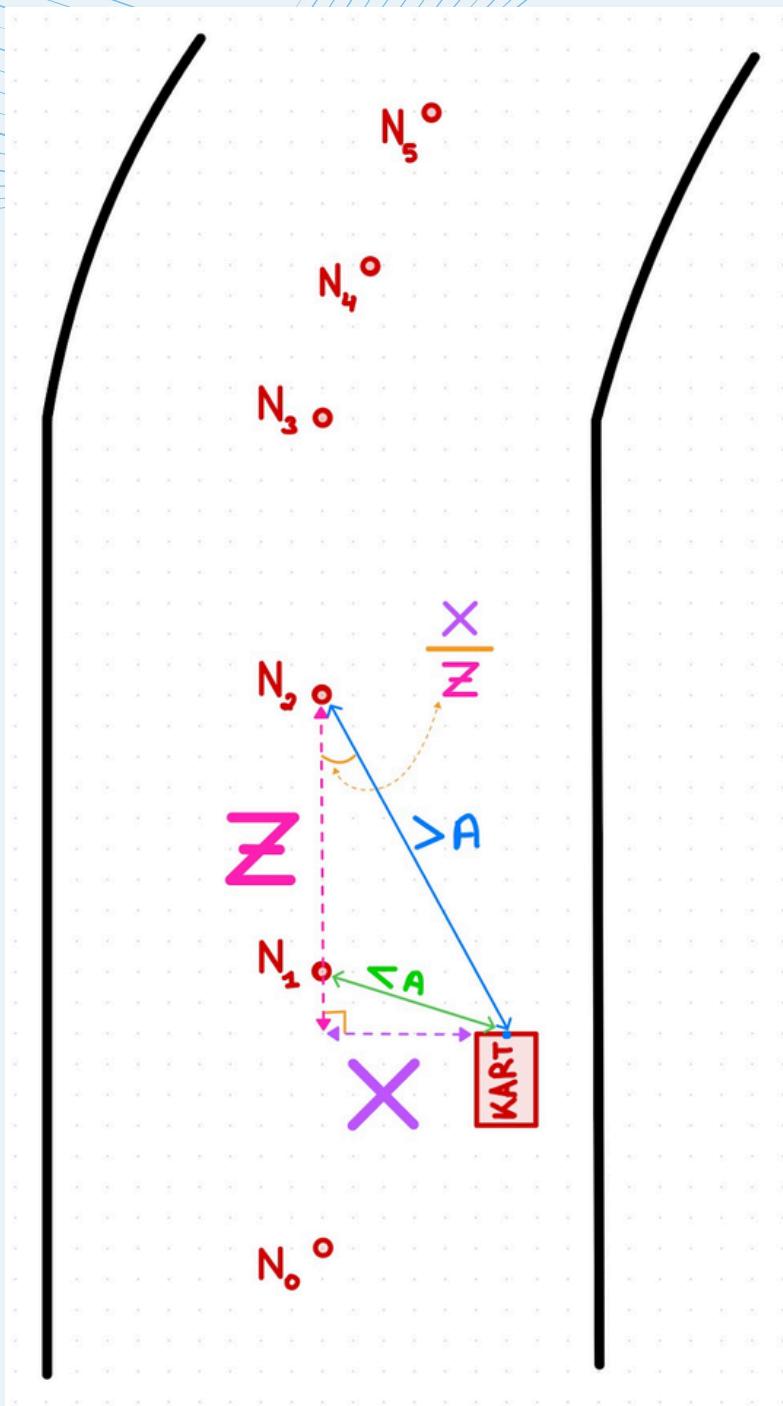
- Entrée : L'écart latéral x et l'écart avant z du point cible
- Sortie : Angle de braquage normalisé et distance longitudinale corrigée

## La méthode `manage_speed`

- Entrée : `distance_down_track velocity`
- Sortie : `accel brake steering`

## La méthode `choose_action`

- Entrée : `observation`
- Sortie : `action{}`



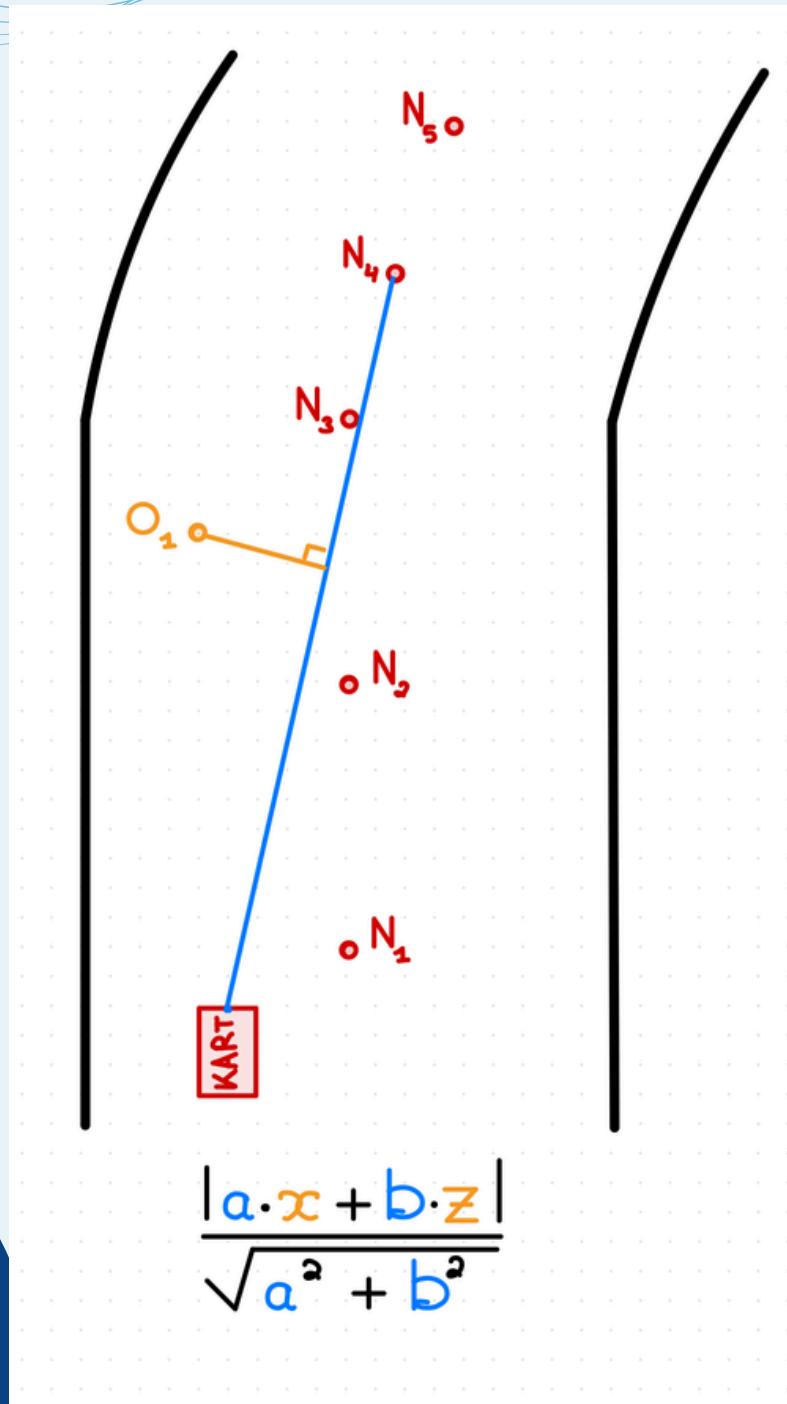
N<sub>i</sub> : Noeud i au centre de la piste

O<sub>i</sub> : Obstacle i stance lookahead\_meters

X : Distance entre notre kart et le centre de l'obstacle

Z : Distance entre le kart et O<sub>i</sub>

# BANANA PILOT



La méthode **position\_track**

- Entrée : paths\_end velocity
- Sortie : l'écart latéral x l'écart avec z

La méthode **edge\_safety**

- Entrée : center\_path\_distance
- Sortie : bool steering accel

La méthode **detect\_banana**

- Entrée : items\_position items\_type
- Sortie : bool steering accel

La méthode **choose\_action**

- Entrée : observation
- Sortie : choose\_action

# NITRO PILOT

## La méthode `detect_nitro`

- Entrée : steering, acceleration, energy, brake, drift
- Sortie : un bool, steer, acel, nitro



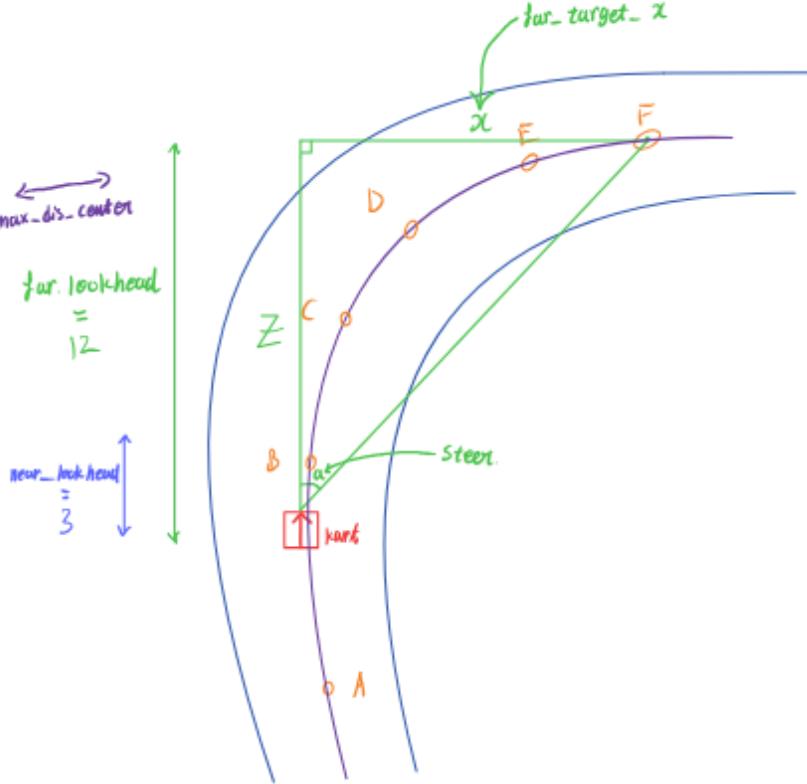
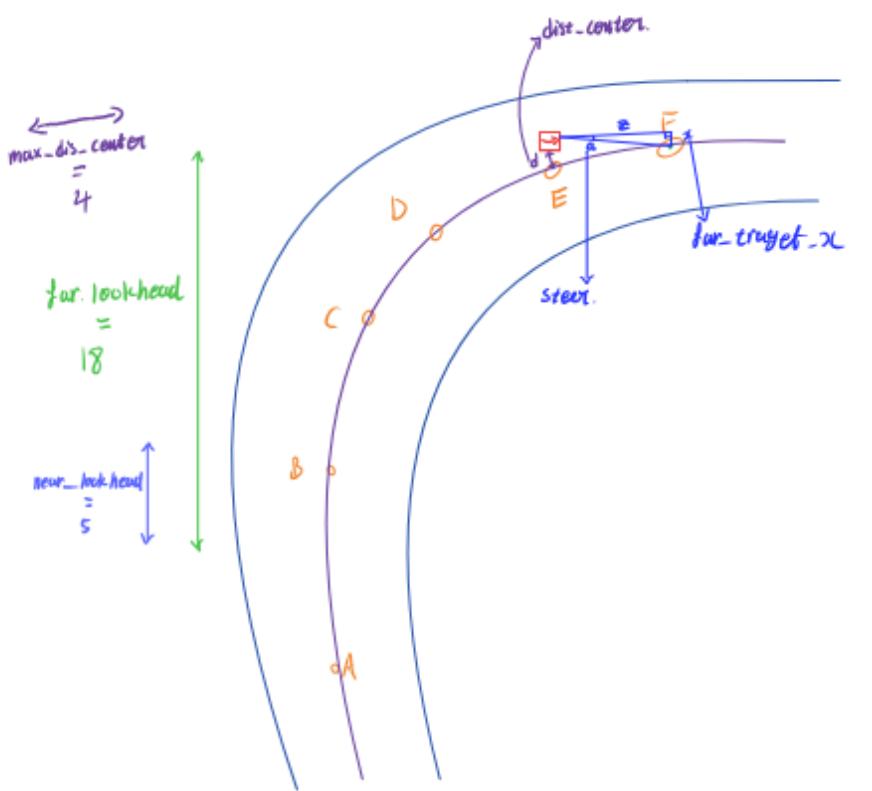
## La méthode `choose_action`

- Entrée : bool (detect nitro)
- Sortie : Retourne une action spécifique à l'utilisation du nitro si le bool de detect nitro est TRUE sinon l'action du wrapper

# DRIFT PILOT

## La méthode `choose_action`

- Entrée : Les observations
- Sortie : dictionnaire d'action complet pour le kart



# MERCI