



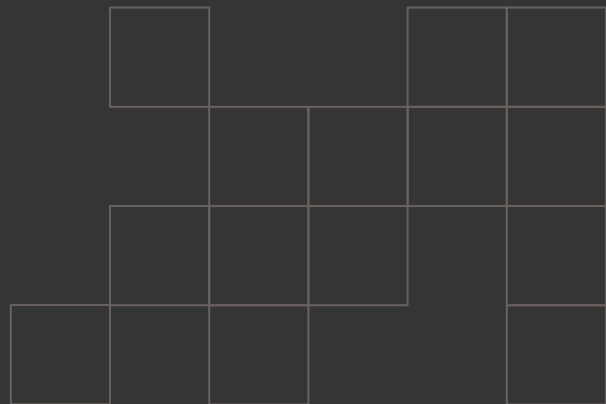
Implémentation et
contrôle d'un agent
intelligent dans
SuperTuxKart

SUPER
**TUX
KART**

Projet IA

Team L'éclair

-Wenxia Wu -Dylan Greedharry -Dmytro Bohov -Hakim Haddouchi -Sanam Khataei -Amine Hadj Daoud



Pilotage du kart

Analyse de
l'environnement



Décision de la
direction



Gestion de
l'accélération



Gestion des
situations
particulières



Conclusion



Gestion du Mouvement – Analyse du Code (Pilot.py)

Pipeline de Contrôle du Kart



1. Observation (obs)

Lecture des données de l'environnement (obs): Vitesse, Position, Piste, Objets



2. Ciblage Noeud

Calcul: $\text{idx} = \text{velocity}/10 + 1 \rightarrow$ Point cible



3. Détection d'Obstacles

Vérification: Bananes sur la trajectoire



4. Calcul d'Angle (err)

Formule: $\text{atan2}(\text{dx}, \text{dz}) \rightarrow$ Angle à braquer



5. Contrôle PD (Braquage)

Calcul: $\text{steer} = \text{PD}(\text{err}) + \text{Ajustements}$



6. Stratégie de Vitesse

Décision: Accélération / Freinage / Nitro



7. Action Finale

Commande: steer, acceleration, brake, nitro



Code par Étape (Pilot.py)

Ciblage & Angle (L14, L35)

```
idx = int((velocity/10)+1.0)
err = math.atan2(target_ctr_x, target_ctr_z)
```

Contrôle PD (L39)

```
ctrl_pd = p_k * err + d_k * (err - self.prev_err)
Correction temps réel pour éviter les oscillations.
```

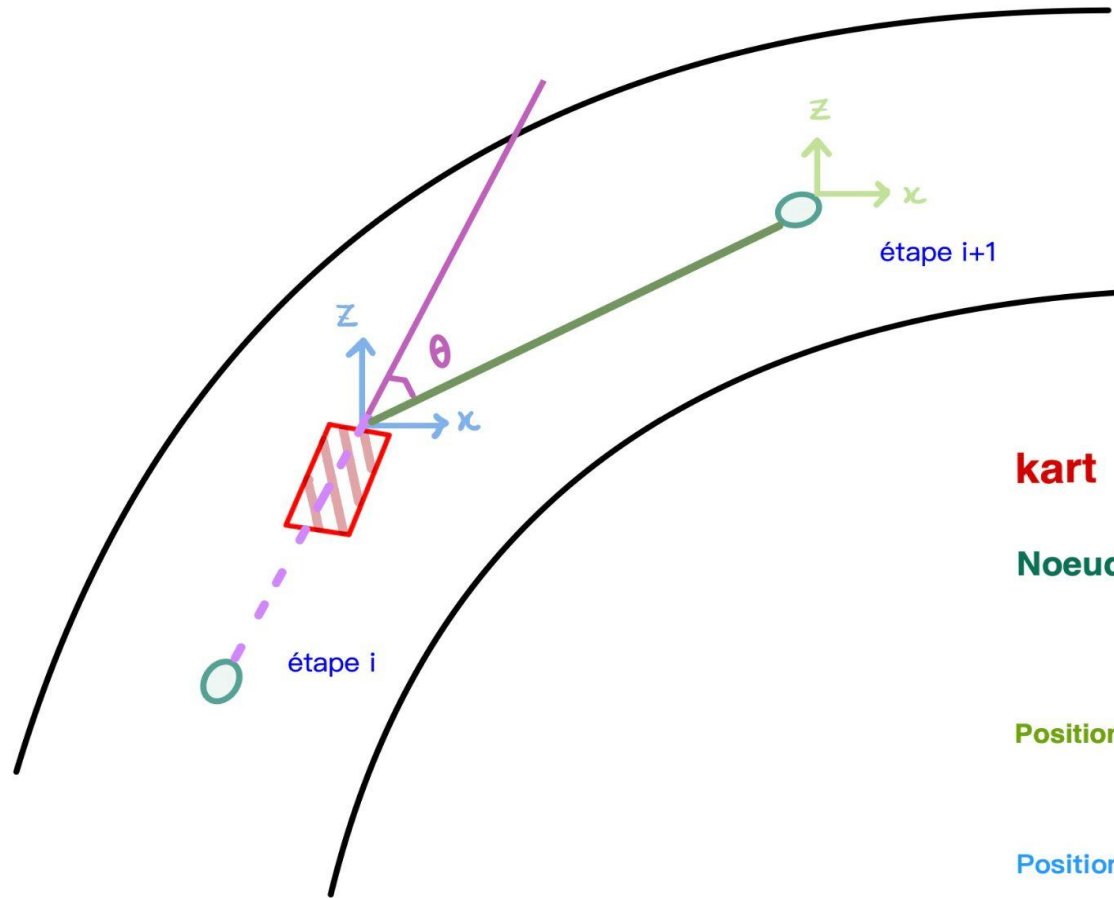
Stratégie Vitesse (L42+)

Brake si $\text{err} > 1.0$; Nitro si $\text{steer} < 0.2$ & $\text{energy} \geq 2$



Variables Clés

- **velocity / speed**: Influence ciblage et stratégie nitro/frein.
- **err (Angle d'erreur)**: Entrée principale du contrôleur de braquage.
- **p_k, d_k (PD Params)**: Définissent la sensibilité et la stabilité.
- **energy**: Niveau d'énergie pour activation nitro.

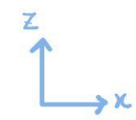
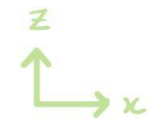


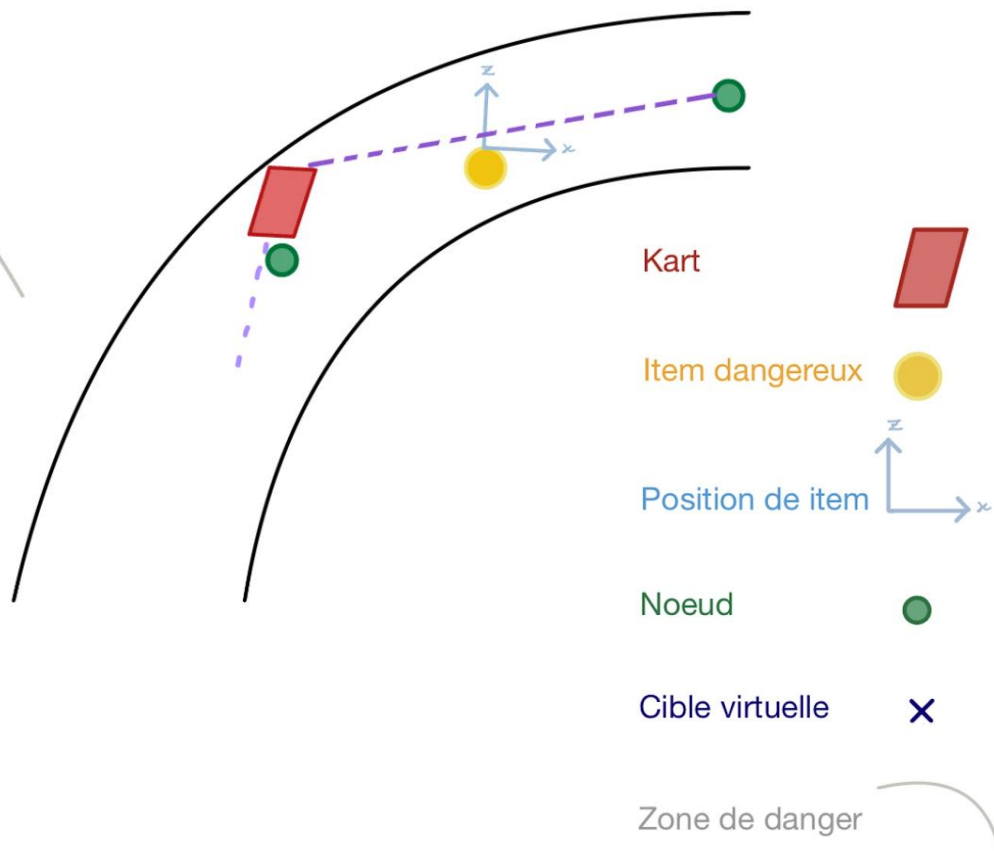
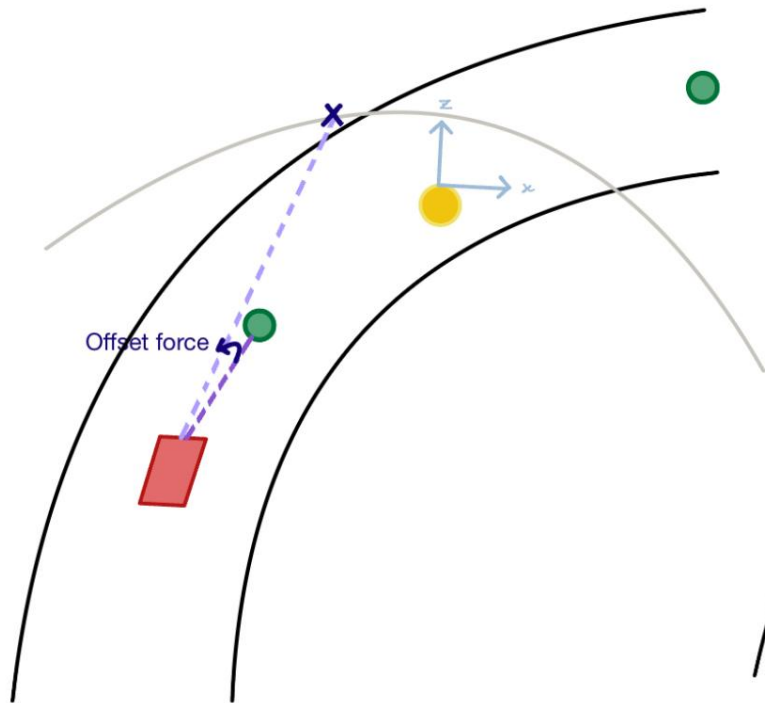
kart

Noeud

Position de noeud

Position de kart







Merci