Implémentation et contrôle d’un agent intelligent dans SuperTuxKart

Projet IA

Team L’eclair

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

Pilotage du kart

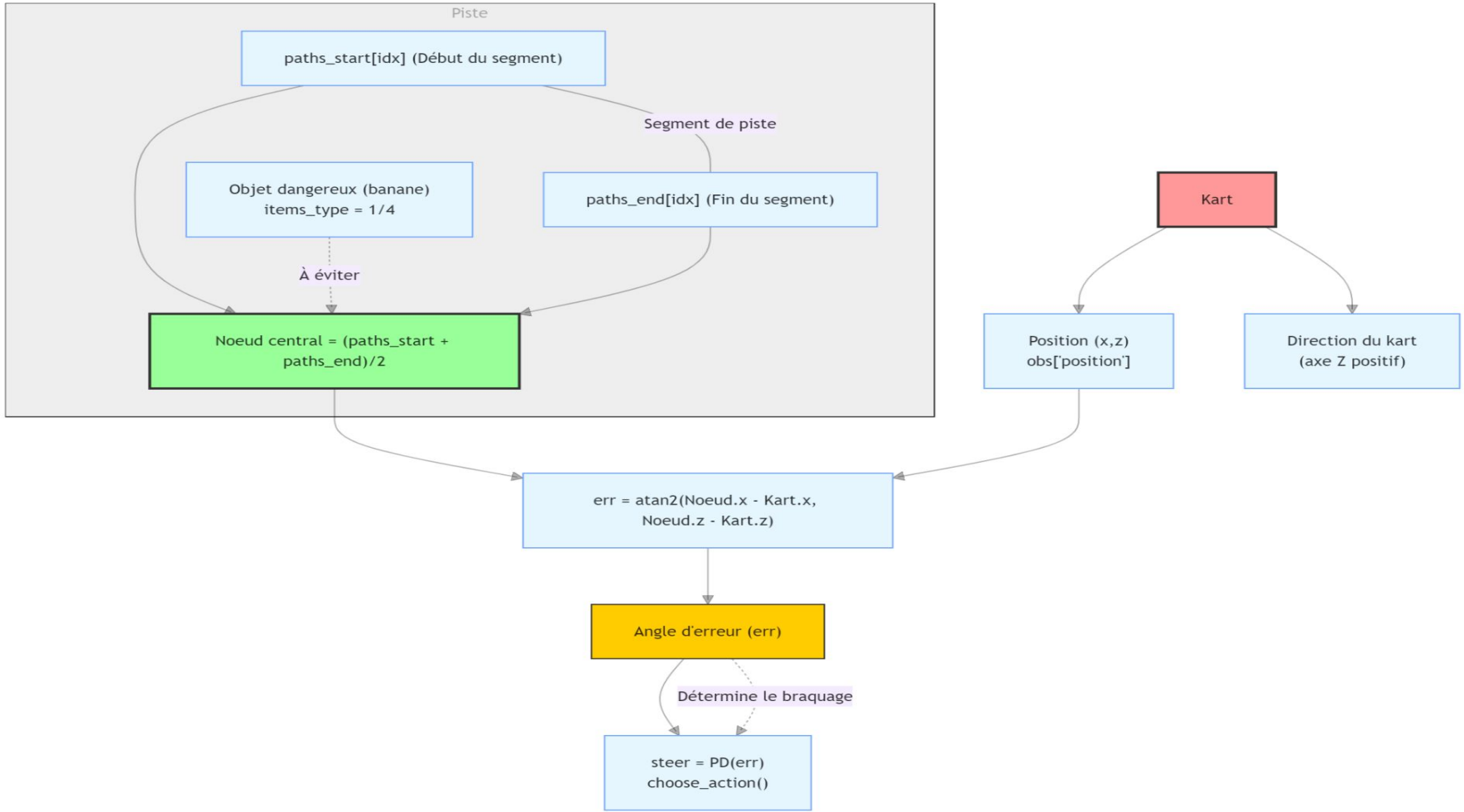
Analyse de l'environnement

Décision de la direction

Gestion du mouvement

Gestion des situations particulières

Conclusion


Piste
paths_start[idx] (Début du segment)
Segment de piste
Objet dangereux (banane)
items_type = 1/4
paths_end[idx] (Fin du segment)
À éviter
Noeud central = (paths_start + paths_end)/2
Kart
Position (x,z)
obs['position']
Direction du kart
(axe Z positif)
err = atan2(Noeud.x - Kart.x, Noeud.z - Kart.z)
Angle d'erreur (err)
Détermine le braquage
steer = PD(err)
choose_action()

Découvrez l'équipe

Décision de la direction

Calcul de la trajectoire

Étape 1: Repère de la position d’un nœud

Étape 2: Calcul de la tangente

Étape 3: Calcul de la dérivée

Repère de la position

```
velocity = obs["velocity"][2]
idx = int((velocity/10)+1.0)
target_ctr_x = (obs["paths_start"][idx][0] + obs["paths_end"][idx][0]) / 2.0
target_ctr_z = (obs["paths_start"][idx][2] + obs["paths_end"][idx][2]) / 2.0
```

- Nœud : Choix par rapport à la vitesse de kart
- X et Z : Les vecteurs pertinents
- Moyen : La moyen entre le début et fin d'une section de piste

Objectif :

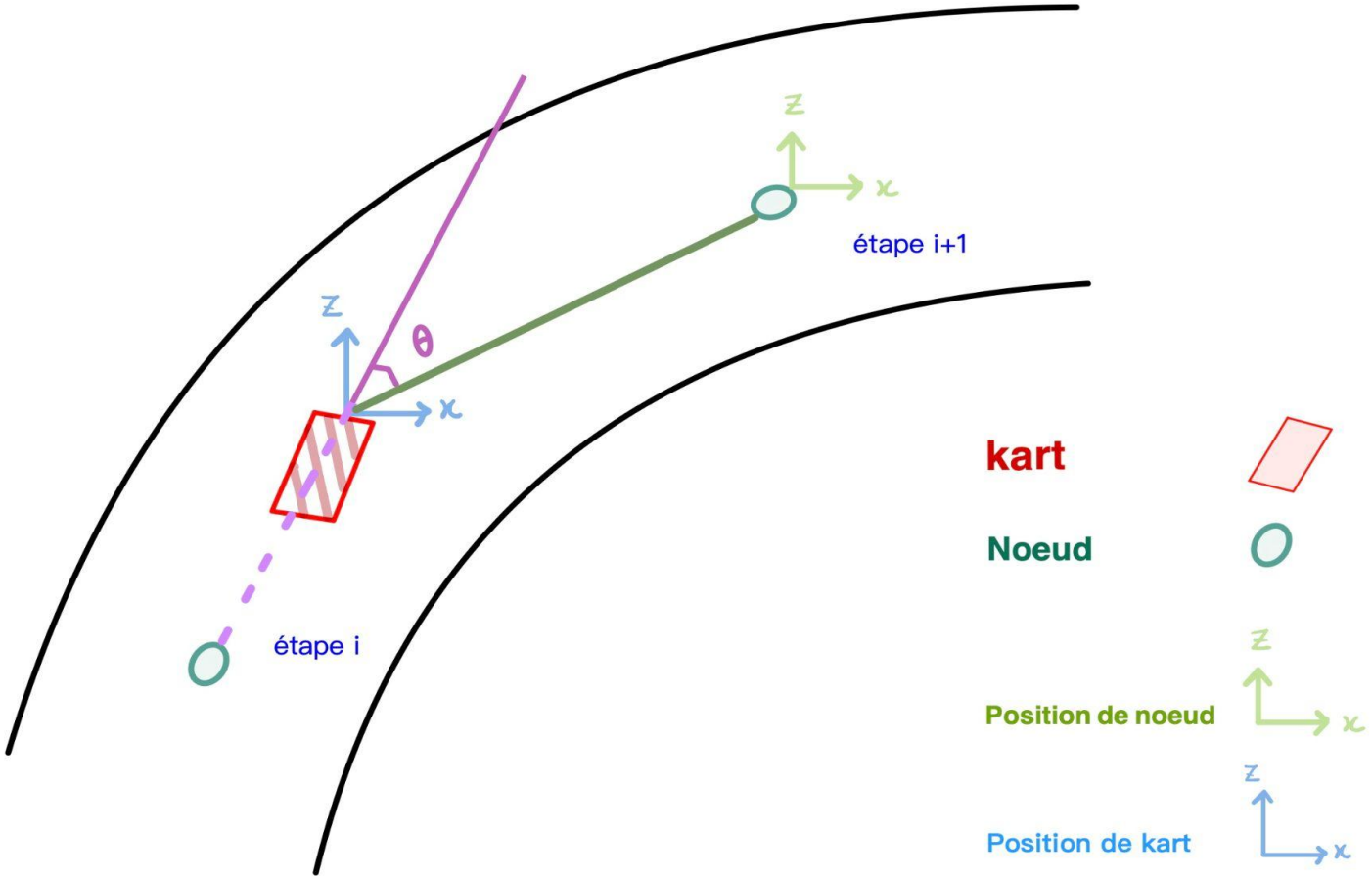
Choisir par rapport auquel nœud nous allons faire le calcul

Calcul de la tangente

```
target_ctr_x += offset_force
err = math.atan2(target_ctr_x, target_ctr_z)
```

Objectif : Trouver l'angle entre la position de kart et le nœud

- offset_force : Plus tard dans la présentation...
- tan(θ) : x / z


z
x
étape i+1
z
θ
x
étape i
kart
Noeud
z
Position de noeud
x
z
Position de kart
x

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: idx = velocity/10 + 1 -> Point cible

3. Détection d'Obstacles
Vérification: Bananes sur la trajectoire

4. Calcul d'Angle (err)
Formule: atan2(dx, dz) -> Angle à braquer

5. Contrôle PD (Braquage)
Calcul: steer = PD(err) + 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 err>1.0; Nitro si steer<0.2 & energy>=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.

Gestion des situations particulières

Esquiver des Bananes / Bubble gums

Étape 1: La Détection

Étape 2: La Réaction

Étape 3: L'Exécution

La Détection (Le "Radar")

- Filtrage d'Objets : Sélection spécifique des bananes (1) et chewing-gums (4).
- Zone de Menace : Analyse d'un couloir de 20 mètres devant le kart.
- Priorité au Proche : Identification de l'obstacle le plus imminent.

```
offset_force = 0.0
items_type = np.array(obs["items_type"])
items_pos = np.array(obs["items_position"])
closest_baditems_dist = 20.0
closest_baditems_x = 0.0
danger = False
for i in range(len(items_type)):
    if items_type[i] in [1, 4]:
        baditems_x = items_pos[i][0]
        baditems_z = items_pos[i][2]
        if 0 < baditems_z < closest_baditems_dist and abs(baditems_x) < 3.5:
            closest_baditems_dist = baditems_z
            closest_baditems_x = baditems_x
            danger = True
```

Objectif :

Expliquer comment l'IA "voit" le danger.

La Réaction (La "Force de Répulsion")

```
if danger:
    avoid_force = (20.0 - closest_baditems_dist) / 2.0
    if closest_baditems_x > 0:
        offset_force = -avoid_force
    else:
        offset_force = avoid_force
```

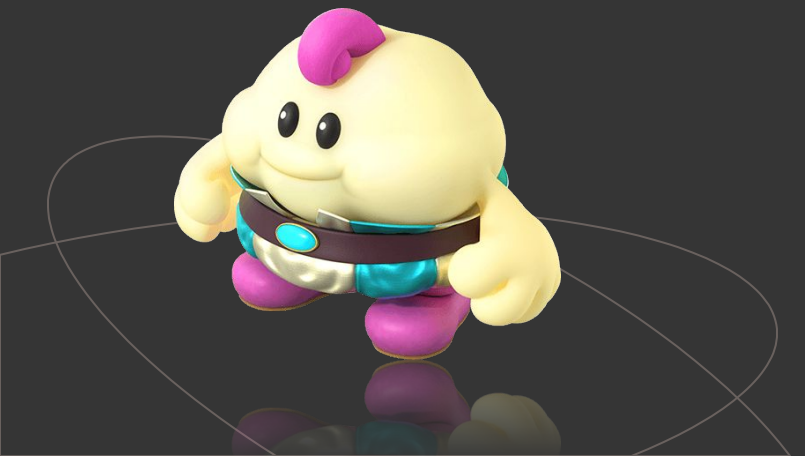
Objectif : Expliquer comment l'IA décide de l'intensité de l'esquive.

- Proportionnalité : Plus l'objet est proche, plus la réaction est vive.
- Vecteur d'Évitement : Calcul d'une force latérale (avoid_force).
- Direction Opposée : Décision automatique de braquer à l'opposé de l'obstacle.

L'Exécution (La "Cible Virtuelle")

Objectif : Expliquer comment l'esquive est intégrée à la conduite.

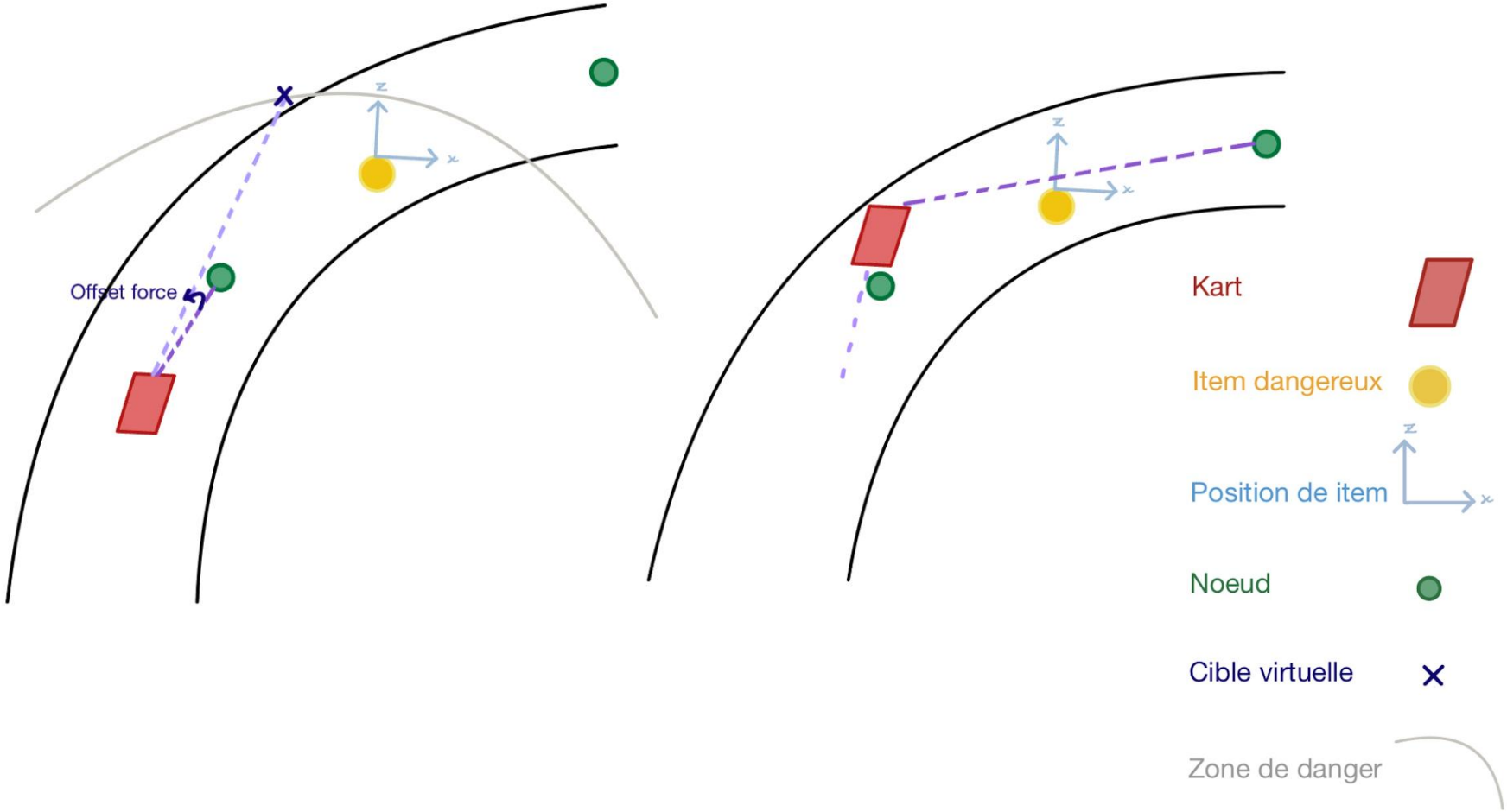
```
target_ctr_x += offset_force
```



Décalage de Cible : On ne change pas l'angle de vue, on déplace la destination.

Cible Virtuelle : Création d'un point de passage sécurisé.

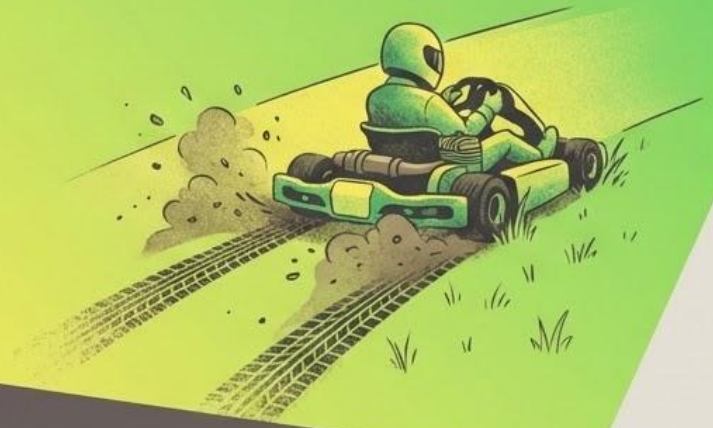
L'idée forte : L'IA ne "fuit" pas l'obstacle, elle recalcule simplement un nouveau centre de route temporaire.


Offset force
Kart
Item dangereux
Position de item
Noeud
Cible virtuelle
Zone de danger

Problèmes actuels

On prend pas bien les virages

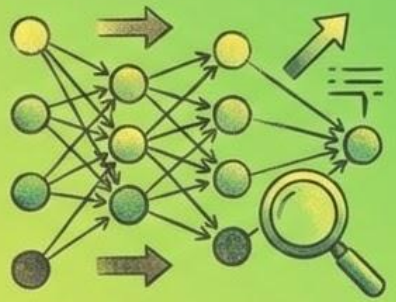
On sort trop souvent de la piste



Futurs objectifs

Implémentation du drift

Optimisation d’hyperparamètres via Optuna



L’éclaire

Demandes pas de questions gros

Any questions? Ask away!

Merci