

Nouvelle fonction path_ajust qui cette fois au lieu d'utiliser la variable center_path_distance utilise les nœuds de la piste devant le kart.

Pour récupérer le 3ème nœud devant le kart:

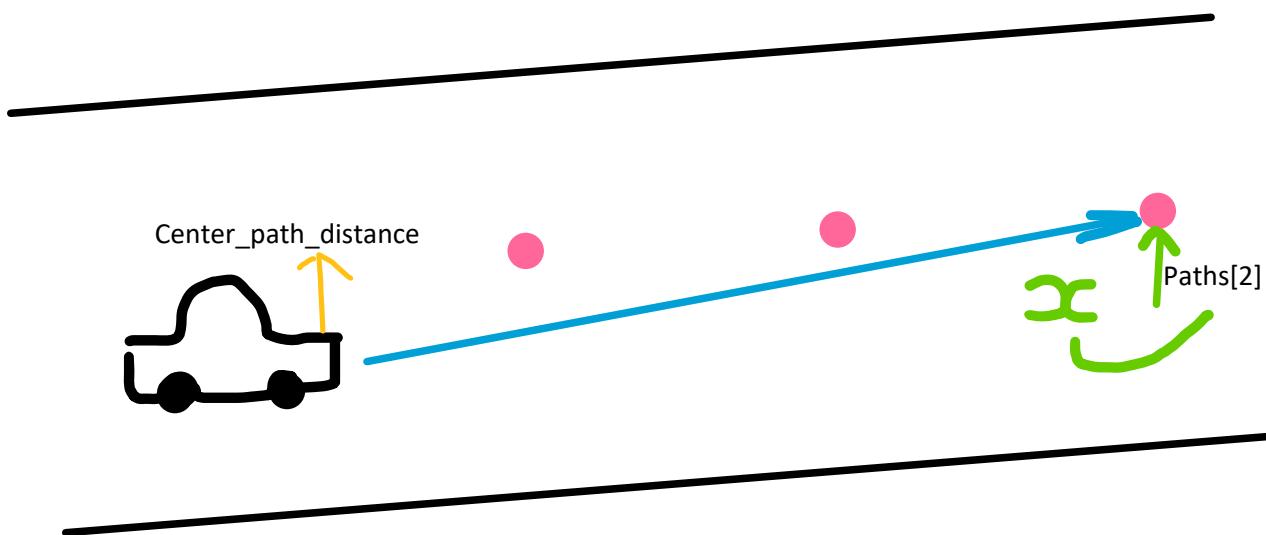
Paths = obs["paths_end"] donne les points devant le kart jusqu'à la fin de la piste
 Paths[2][0] récupère le troisième point avec [2] et sa coordonée sur x avec [0]

s'il y a moins de deux points devant nous c'est qu'on est presque à la ligne d'arrivée du coup ce n'est pas intéressant de regarder et on return

Sinon, c'est le même principe que la fonction d'avant, j'accepte une certaine distance self.dist d'écart avec le x de ce deuxième point devant nous; Si je vais au delà de cet écart acceptable, alors j'ajuste ma trajectoire en modifiant le steer

`s = max(-1.0, min(1.0, s))`

Force le steer à être compris entre -1 et 1 pour la cohérence, puisque la fonction path_ajust fait des additions sur le steer, celui-ci doit être borné

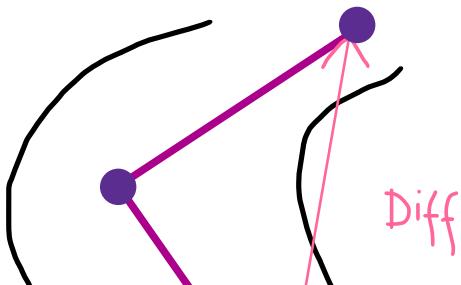


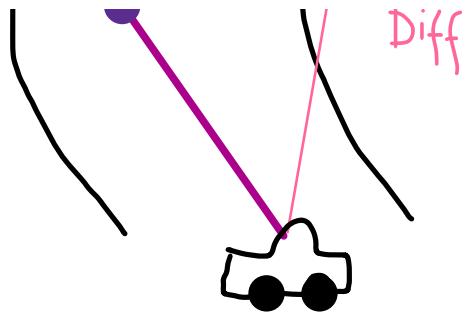
Premier prototype de drifting :

Reprise du code de Bilal ; et ajout de conditions :

Si le code reconnaît un virage serré avec la variable écart (écart entre les segments devant le kart), si cet écart est grand, alors c'est un virage serré ; dans ce cas-là on drift et on steer , mais de quel sens ?

Diff est la variable qui donne la différence des vecteurs formé par les deux segments :





```
segdirection = end - start      # vecteur du segment
diff = segdirection - front    # comparaison avec direction "front"
ecart = float(np.linalg.norm(diff)) # distance
```

Segdirection c'est le vecteur différence entre les deux segments et diff c'est les coordonnées de segdirection en fonction du kart

```
if ecart >= self.ecartgrand: #virage serré
    s = act["steer"]
    if (diff[0] < 0): #virage a droite si diff x negatif
        s = -0.6 #0.6 a tester
    act["steer"] = s

    elif (diff[0] > 0): #virage a gauche
        s = 0.6
        act["steer"] = s
    act["drift"] = True
```

Et on drift à la fin après avoir tourné le volant pour drift dans le bon sens