Stratégies des différentes classes d'Agents

Rouge = informations à vérifier/tester

Sommaire :

AgentCenter

Agent chargé de suivre le centre de la piste.

Stratégies

A chaque instant, on observe le 3e (dépend de path_lookahead) noeud devant nous. Si ce nœud est trop à gauche ou trop à droite par rapport à notre kart (seuil défini par une variable DIST), on multiplie la coordonnée X de ce nœud par une variable AJUST définie dans le fichier de configuration.

Pseudo-codes

Une méthode Ajustement du path(action, obs).
Ainsi que la méthode choose_action de base corrigée.

Ajustement du path(action, obs) :

Récupération du steer actuel de notre kart
Récupération du 3e (path_lookahead) noeud devant notre kart
Si le prochain noeud est loin et qu'on est à peu près au centre de la piste :
    Steer réinitialisé à 0
Sinon, si la coordonnée X du noeud est supérieure à DIST :
    Steer = AJUST * coordonnée X du noeud
Steer borné dans le dictionnaire d'actions entre -1 et 1
Retourne le dictionnaire d'actions

AgentSpeed

Agent qui régule la vitesse en fonction de la courbature (les virages et lignes droites) de la piste.

Stratégies

La régulation de la vitesse se fait en deux étapes : une analyse des prochains segments de la piste, et une réaction en fonction de l'analyse obtenue.
La méthode d'analyse observe les 3 (path_lookahead) prochains nœuds et fait des mesures en fonction de leur distance par rapport à notre kart et l'angle d'écart. Si l'un de ces nœuds réunit les bonnes conditions, on considère qu'on est sur un virage serré. On renvoie un booléen virage_serré qui décrit le segment sur lequel on est.
La méthode de réaction récupère le précédent booléen et agit en conséquence. Si on est sur une ligne droite, on initialise l'accélération à une valeur définie par le fichier de configuration. Si on est sur un virage serré, on récupère la valeur max du steer possible selon notre vitesse actuelle (dans le dictionnaire d'observations, max_steer_angle ou MSA). Selon le MSA actuel, on accélère ou on décélère. Si on est sur une pente, on donne un léger coup de boost à l'accélération.

Pseudo-codes

2 méthodes :

- Analyse(obs)
- Réaction(virage_serré, action, obs)

Ainsi que la méthode choose_action de base corrigée.

Analyse(obs) :

Initialisation de virage_serré à False
Récupération du nombre de segments entre le maximum entre path_lookahead et le nombre de noeuds restants

Pour chaque i dans la range du nombre de segments :
Récupération du segment en faisant la différence entre paths_end[i] et paths_start[i]
Calcul du vecteur de la différence entre le segment et notre kart
Calcul de la longueur de ce vecteur
Calcul de la longueur du segment

Si l'écart directionnel est plus grand qu'un seuil déterminé et que la longueur du segment est inférieure à un autre seuil déterminé :
virage_serré initialisé à True
Renvoie virage_serré

Réaction(virage_serré, action, obs) :
Récupération du MSA
Si virage_serré est False :
L'accélération est initialisée à la valeur pour les lignes droites
Retourne le dictionnaire d'actions
Si MSA est inférieur à un certain seuil :
On réduit l'accélération
Sinon, si MSA est supérieur à un certain seuil :
On augmente l'accélération
Si la coordonnée Y est supérieure à un certain seuil :
On est sur une pente, on doit accélérer un peu plus
Retourne le dictionnaire d'actions

AgentObstacles

Agent chargé de se diriger vers les bonus et d'éviter les obstacles.

Stratégies

A chaque instant, vérifier le type des prochains objets devant nous.
Si ce sont des obstacles, appeler une méthode qui se charge d'éviter ces obstacles.
Si ce sont des bonus ou des nitros, appeler une méthode qui se charge de se diriger vers ces objets.

Éviter l'obstacle :
On note l'index de l'obstacle qu'on observe.
Si l'item est trop loin et peut être ignoré, ou si on l'a déjà passé (donc il a déjà été géré), on supprime la notation de l'index et on renvoie le dictionnaire d'actions sans rien changer.
Si on a un shield, on renvoie le dictionnaire d'actions sans rien changer.
Sinon, on récupère le steer actuel et on le compare aux coordonnées X du prochain obstacle. Si on se dirige actuellement vers l'obstacle, on réduit ou on augmente le steer d'une valeur fixe (peut être un hyperparamètre). On renvoie ensuite le dictionnaire d'actions modifié.

Se diriger vers les bonus :
On note l'index du bonus qu'on observe.
Si l'item est hors de portée, on supprime la notation de l'index et on renvoie le dictionnaire d'actions sans rien changer.
Si l'item est trop proche de nous en coordonnée z, on abandonne l'idée de se diriger vers lui, on supprime la notation de l'index et on renvoie le dictionnaire d'actions sans rien changer.
Sinon, on récupère le steer actuel et on le compare aux coordonnées X du prochain bonus. Si on se dirige actuellement vers le bonus, on

ne change rien et on renvoie le dictionnaire d'actions. Sinon, la coordonnée X du bonus devient notre steer. On renvoie le dictionnaire d'actions modifié.

Pseudo-codes

3 méthodes :

- Observation du prochain item(obs, action)
- Éviter obstacle(obs, action, index)
- Dirige vers bonus(obs, action, index)

Ainsi que la méthode choose_action de base corrigée.

Observation du prochain item(obs, action) :

tableau des index bonus = [tous les index des bonus dans obs[“items_type”]
#les bonus correspondent à 0, 2, 3
tableau des index obstacles = [tous les index des obstacles dans obs[“items_type”]
#les obstacles correspondent à 1, 4
pour chaque i dans le tableau des bonus :
appel à Dirige vers bonus(obs, action, i)
pour chaque i dans le tableau des obstacles :
appel à Éviter obstacle(obs, action, i)
Renvoie le dictionnaire d'actions

NB : On appelle d'abord la méthode pour les bonus avant la méthode pour les obstacles pour que les obstacles soient la dernière modification faite au dictionnaire d'actions, donc la plus importante

Éviter obstacle(obs, action, index) :

Si on a aucun obstacle ciblé :
obstacle ciblé = index
Récupération du vecteur de cet obstacle

Si l'obstacle est très proche de nous ou si au contraire il

est trop loin :
obstacle ciblé = None

Si on a un obstacle ciblé qui correspond bien à index :
Si on a un shield bubblegum et qu'il reste activé assez de temps :
On renvoie le dictionnaire d'actions sans modification
Si l'obstacle est à notre gauche, on augmente le steer
Si l'obstacle est à notre droite, on réduit le steer
Renvoie le dictionnaire d'actions

NB : Garder l'obstacle ciblé en attribut nous permet d'éviter les obstacles un par un et ne pas avoir de conflit dans le dictionnaire si on a par exemple deux bananes côte à côte (l'agent ne tournera pas à gauche et à droite en même temps).

Dirige vers bonus(obs, action, index) :
Une optimisation de cette méthode est peut-être possible
Si on n'a aucun item ciblé :
item ciblé = index
Récupération du vecteur de cet item

Si l'item est très proche de nous ou si au contraire il est trop loin :
item ciblé = None

Si on a un item ciblé qui correspond bien à index :
Notre steer devient la coordonnée X du vecteur de l'item
Renvoie le dictionnaire d'actions

NB : De la même manière que pour les obstacles, garder l'index de l'item ciblé permet de ne pas viser deux items en même temps pour être sûr d'en avoir au moins 1.

AgentItems

Agent qui gère l'utilisation des bonus.

Stratégies

A chaque instant, vérifier si on a un item à disposition.
Si on est très proche d'une boîte à items, en général, activer l'item possédé (pas à chaque fois !).
Actions pour chaque item :

Bubblegum
Si on est très proche d'un obstacle, activer le shield et suivre le chemin de base sans dévier l'obstacle.
Si une balle de basket ou une bombe gâteau nous fonce dessus (si on a l'info), activer le shield.

Cake
Si on n'est pas premier et si on peut l'activer à n'importe quelle distance du joueur devant nous, alors activer directement.
Sinon, attendre d'atteindre la distance minimale et activer.

Bowling ball
S'il y a un ennemi en face de nous et que la variation de x par rapport à notre kart n'est pas très élevé, diriger légèrement le kart vers lui et activer.
Vérifier si on peut lancer la balle derrière nous (dans ce cas là on devra s'orienter à l'envers).

Zipper
Si on est sur une ligne droite et qu'on n'est pas en train d'éviter un obstacle, activer.
Plus tard, optimiser ça avec des raccourcis ?

Plunger
Si on a un virage devant nous à la distance d'activation, activer.

Si on a un joueur devant nous à la distance d'activation, diriger légèrement le kart vers le joueur en face et activer.

Switch

Si on a un avantage (boîte d'item ou nitro) devant nous, NE PAS ACTIVER.
Sinon, activer directement.
Peut-être attendre qu'un joueur ennemi soit en face d'un avantage pour qu'il ne puisse pas l'esquiver ?

Swatter

Si un ennemi est à un rayon de moins de 10m de nous, activer.

Rubberball

Si on n'est pas premier, activer directement.

Parachute

Si on n'est pas premier, activer directement.

Pseudo-codes

Une méthode Observation de l'item actuel(obs, action).
Ainsi que la méthode choose_action de base corrigée.

Observation item actuel(obs, action) :

Récupération de l'item qu'on possède

Si notre item vaut 0 ou 10 :
#0 : aucun item, 10 : item non utilisable (ANVIL)
Retourne le dictionnaire d'actions

Si notre item est un BUBBLEGUM (1) :
Si on est en train d'éviter un obstacle :
On arrête de l'éviter
Activation de l'item
Retourne le dictionnaire d'actions

Si on est visé par un malus ennemi :
Activation de l'item
Retourne le dictionnaire d'actions
Si on se dirige vers un bonus, que ce bonus est une boîte d'items et qu'il est très proche de nous :
Activation de l'item
Retourne le dictionnaire d'actions

Si notre item est un CAKE (2) :
Récupération de la position de tous les kart ennemis
Si l'un d'eux est à une distance devant nous assez proche pour être touché par la bombe :
Activation de l'item
Retourne le dictionnaire d'actions
Si on se dirige vers un bonus, que ce bonus est une boîte d'items et qu'il est très proche de nous :
Activation de l'item
Retourne le dictionnaire d'actions

Si notre item est une BOWLING BALL (3) :
Récupération de la position de tous les karts ennemis
Si l'un d'entre eux est à une distance devant nous assez proche et centrée par rapport à notre kart :
Se diriger légèrement vers ce kart
Activation de l'item
Retourne le dictionnaire d'actions
Voir si on peut le lancer derrière
Si on se dirige vers un bonus, que ce bonus est une boîte d'items et qu'il est très proche de nous :
Activation de l'item
Retourne le dictionnaire d'actions

Si notre item est un ZIPPER (4) :
Appel de la fonction d'analyse d'AgentSpeed
Si on est sur une ligne droite et qu'on n'est pas en train d'éviter un obstacle :
Activation de l'item

Retourne le dictionnaire d'actions

Viser les raccourcis si on a un ZIPPER

Si notre item est un PLUNGER (5) :

Récupération de la position de tous les karts ennemis

Si l'un d'entre eux est à une distance devant nous assez proche et centrée par rapport à notre kart :

Se diriger légèrement vers ce kart

Activation de l'item

Retourne le dictionnaire d'actions

Appel de la fonction d'analyse d'AgentSpeed

Si on est sur un virage et assez proche du mur en face :

Activation de l'item

Retourne le dictionnaire d'actions

Si on se dirige vers un bonus, que ce bonus est une boîte d'items et qu'il est très proche de nous :

Activation de l'item

Retourne le dictionnaire d'actions

Si notre item est un SWITCH (6) :

Si on se dirige vers un bonus :

action["fire"] reste à 0

Retourne le dictionnaire d'actions

Attendre qu'un ennemi soit devant un bonus pour esquiver

Activation de l'item

Retourne le dictionnaire d'actions

Si notre item est un SWATTER (7) :

Récupération de la position de tous les karts ennemis

Si l'un des karts ennemis est à la distance d'activation de la tapette à mouche :

Activation de l'item

Retourne le dictionnaire d'actions

Si on se dirige vers un bonus, que ce bonus est une boîte d'items et qu'il est très proche de nous :

Activation de l'item

Retourne le dictionnaire d'actions

Si notre item est une RUBBERBALL (8) :

Récupération de la position du premier kart (sans nous compter)

Si le premier kart (sans nous compter) est devant nous :

Activation de l'item

Retourne le dictionnaire d'actions

Si on se dirige vers un bonus, que ce bonus est une boîte d'items et qu'il est très proche de nous :

Activation de l'item

Retourne le dictionnaire d'actions

Si notre item est un PARACHUTE (9) :

Récupération de la position du premier kart (sans nous compter)

Si le premier kart (sans nous compter) est devant nous :

Activation de l'item

Retourne le dictionnaire d'actions

Si on se dirige vers un bonus, que ce bonus est une boîte d'items et qu'il est très proche de nous :

Activation de l'item

Retourne le dictionnaire d'actions

action[“fire”] reste à 0

Retourne le dictionnaire d'actions

AgentRescue

Agent chargé de reculer lorsqu'on est coincé sur la piste (face à un mur).

Stratégies

Vérifier qu'on est bloqué à chaque instant en comparant notre distance par rapport au début de la piste à la dernière distance enregistrée (qu'on actualise à chaque fois qu'on réalise qu'on n'est pas bloqué).
Si on est bloqué, appeler une méthode qui s'occupe de reculer pendant un certain nombre de st
eps. Pendant qu'on recule, toutes les autres actions sont annulées.
Un nouvel attribut booléen est chargé de retenir si le kart doit reculer ou pas pour éviter les conflits avec les autres méthodes.

3 variables dans le fichier de configuration pour cette classe :

- MIN_PROGRESS_THRESHOLD : le seuil d'écart de distance à partir duquel on considère qu'on est immobile ou bloqué.
- BLOCK_COUNTER_THRESHOLD : le nombre de ticks à partir duquel on considère être bloqué.
- UNBLOCK_STEPS_DEFAULT : le nombre de steps par défaut pendant lequel le kart doit reculer.

Pseudo-codes

2 méthodes :

- Est bloqué(obs)
- Action de déblocage(action)

Ainsi que la méthode choose_action de base corrigée.

Est bloqué(obs) :

Récupération de la distance par rapport au début de la piste
Récupération de l'**attachment**

Si la dernière distance enregistrée est inexistante :

La dernière distance enregistrée devient notre distance actuelle par rapport au début de la piste

Si la différence entre la distance actuelle et la dernière distance enregistrée est inférieure à MIN_PROGRESS_THRESHOLD et qu'on est à plus de 5 unités du début de la piste et que le kart n'est pas en train de sauter :

Augmentation du compteur de blocage

Sinon :

Compteur de blocage réinitialiser à 0

La dernière distance enregistrée devient la distance actuelle

Action de déblocage(action) :

Si le nombre de pas de reculs restant est positif :

Nombre de pas de reculs restant -= 1

Dictionnaire d'action défini pour reculer

Retourne le dictionnaire d'actions

Sinon

On retire le statut "doit reculer" du kart

Retourne le dictionnaire d'actions

Choose_action(obs) :

Appel de la fonction **Est bloqué**

Récupération du dictionnaire d'actions par **Choose action** du wrapper dont on hérite

Si le compteur de tics durant lequel on est bloqué est supérieur à BLOCK_COUNTER_THRESHOLD :

Le kart est au statut "doit reculer"

Le nombre de steps durant le kart doit reculer est initialisé à UNBLOCK_STEPS_DEFAULT

Si le kart est au statut "doit reculer" :

Appel de la fonction **Action de déblocage**

Retourne le dictionnaire d'actions

AgentDrift

Agent qui améliore la performance dans les virages en dérapant.

Stratégies

A remplir

Pseudo-codes

A remplir