Variables d'observation

Dans un script testagent.py (qui est une copie du script single_track_race_display.py), on a ajouté une fonction affichage_variables pour analyser avec les variables observées en fonction du temps.

Ces observations sont également dans les attributs de notre classe Agent (self.obs), on pourra donc y accéder rapidement afin de décider de la prochaine action à faire.

```
def affichage_variables(action, obs) :
    """Affichage des variables toutes les secondes"""
    global dernier_affichage
    maintenant = time.time()
    if maintenant - dernier_affichage >= INTERVALLE:
        dernier_affichage = maintenant
        for i in range(len(obs["0"]["paths_start"])) :
            print(obs["0"]["paths_start"][i])
```

Par exemple, ici on observait la variable paths_start pour déterminer son contenu au fil du temps

On a pu récupérer les clés du dictionnaire contenant toutes les valeurs observées au fil du temps, qui sont dans l'ordre :

powerup
Type : int
Représente l'item/power-up collecté par notre kart, sous la forme d'un entier. Ce powerup peut être utilisé pour nous donner un avantage, ou donner un désavantage aux adversaires. Le type de power-up change selon l'entier :

Valeur en entier	**Power-up**
0	NOTHING (aucun powerup)
1	BUBBLEGUM
2	CAKE
3	BOWLING
4	ZIPPER
5	PLUNGER
6	SWITCH
7	SWATTER
8	RUBBERBALL
9	PARACHUTE
10	ANVIL

Description des power-ups :

BUBBLEGUM

Octroie un bouclier qui protège notre kart pendant quelques secondes des bananes et des malus venant des adversaires. A la fin du compteur, le bubblegum se pose au sol comme une tâche et ralentit tous ceux qui roulent dessus. Si on roule sur une banane, le bouclier prend fin directement.

CAKE

Se lance sans visée sur le kart ennemi le plus proche et lui explose dessus et sur les karts proches.

BOWLING

Lance une boule de bowling (équivalent à la carapace sur Mario Kart) qui étourdit un instant tous les adversaires touchés. La boule rebondit sur les murs. On peut être touché par la boule même si on en est le lanceur.

ZIPPER

Équivalent des champignons rouges dans Mario Kart, donne un bonus de vitesse pendant un instant.

PLUNGER

Ventouse qui, à l'activation, fonce tout droit ou en arrière et se colle à un mur ou à un kart adverse (à une certaine distance maximale à déterminer). Lorsqu'elle arrive à se coller à quelque chose, on est tiré vers l'objet collé, ce qui nous donne de la vitesse bonus. Si on la lance en arrière, elle bloque la vue d'un adversaire derrière nous (équivalent des Bloups dans Mario Kart, mais n'a aucune utilité ici).

SWITCH

Echange pendant quelques instants les items présents sur la map : les boîtes mystère deviennent des bananes (et inversement), les nitro deviennent des chewing-gum (et inversement).

SWATTER

Tapette à mouche qui reste dans nos mains une dizaine de secondes. Lorsqu'un kart ennemi s'approche trop de notre kart alors qu'on tient cet objet, le kart ennemi est frappé et étourdit pendant quelques secondes. Permet aussi de se débarrasser des Cakes et des Parachutes.

RUBBERBALL

Équivalent de la carapace bleue sur Mario Kart, poursuit le kart en première position en rebondissant, écrase et ralentit les karts qu'elle touche sur son passage. On peut s'en débarrasser en lançant une boule de bowling ou un Cake dessus.

PARACHUTE

A l'activation, ralentit tous les karts devant le nôtre pendant quelques instants. Plus le kart ennemi va vite, plus le ralentissement est important. L'effet du parachute continue tant que le kart n'a pas atteint une certaine vitesse, ou tant qu'on ne

s'est pas cogné à un mur (à vérifier). Il vaut mieux parfois ralentir son kart volontairement pour se débarrasser de l'effet plus vite.

ANVIL

Nous ralentit d'un coup pendant environ deux secondes. Ce "power-up" qui n'en est pas vraiment un n'est pas ramassable mais plutôt un effet qu'on peut obtenir en roulant sur une banane.

NB : En roulant sur une banane, on peut subir les effets de l'ancre, du parachute ou bien d'une bombe qui nous projette dans les airs et nous étourdit quelques secondes.

attachment

Type : int

Définit ce qui est "attaché" à notre kart, c'est-à-dire un item dont notre kart subit l'avantage ou le désavantage. Cette information peut être utile par exemple dans le cas du Shield du Bubblegum qui, s'il est actif, peut nous permettre de suivre la piste sans s'occuper des obstacles (bananes) par exemple. La variable attachment prend ces valeurs :

Valeur en entier	**attachment**
0	PARACHUTE
1	ANVIL
2	BOMB
3	SWATTER
6	BUBBLEGUM_SHIELD
9	NOTHING (aucun attachment)

Description des attachment :

PARACHUTE

Notre kart est ralenti jusqu'à atteindre une certaine vitesse ou après un certain temps.

ANVIL

Notre kart est ralenti d'un coup pendant environ deux secondes.

BOMB

Notre kart est projeté en l'air et étourdi pendant environ 2 secondes.

SWATTER

Notre kart est étourdi puis ralenti pendant quelques secondes.

BUBBLEGUM_SHIELD

Lorsqu'on en est équipé, on est protégé de tous les malus (banane, tapette à mouche) pendant un certain délai.

attachment_time_left

Type : numpy.ndarray (float)
Tableau avec une seule valeur.
Donne le temps restant jusqu'à la fin de l'effet de l'attachment, s'il y en a un. Donne 0 sinon.

max_steer_angle

Type : numpy.ndarray (float)
Tableau avec une seule valeur.
Donne l'angle de manœuvre du volant maximal selon la vitesse actuelle.

=========Donner un tableau pour comparer max_steer_angle par=======
==========rapport à la vitesse du kart==========

energy

Type : numpy.ndarray (int)
Tableau avec une seule valeur.

Donne la quantité restante d'énergie (Nitro) que notre kart a amassée, qu'on peut utiliser pour augmenter pendant un court instant la vitesse de notre kart.

==========Détailler l'utilisation de nitro==========

jumping
Type : int
Renvoie 1 si le kart est en train de sauter, 0 sinon.

distance_down_track
Type : numpy.ndarray (float)
Tableau avec une seule valeur.
Donne la distance parcourue sur le tour à partir de la ligne de départ.
C'est (supposément) la variable utilisée pour déterminer qui est le plus avancé dans la course, et (encore supposément) si on a fini la course ou non.

velocity
Type : numpy.ndarray (tableau à une dimension de taille 3, 3 float)
Vecteur 3D de vitesse de notre kart (x, y, z).
Ce vecteur est dans le référentiel du kart.
(x : gauche du kart, y : haut du kart, z : en face du kart)

front
Type : numpy.ndarray (tableau à une dimension de taille 3, 3 float)
Vecteur 3D du devant de notre kart
Ce vecteur est dans le référentiel de la piste.

center_path_distance
Type : numpy.ndarray (float)
Tableau avec une seule valeur.
Donne la distance de notre kart par rapport au centre de la piste (positif si on est à gauche, négatif si on est à droite).
C'est l'une des variables qu'on va utiliser pour aider notre kart à rester au milieu de la piste.

center_path

Type : numpy.ndarray (tableau à une dimension de taille 3, 3 float)
Vecteur 3D vers le centre de la piste.
Ce vecteur est dans le référentiel de notre kart.

items_position

Type : tuple
Tuple de vecteurs 3D de la position de tous les items sur la piste. Les vecteurs sont dans le référentiel de notre kart, ils changent donc au fil de l'avancée de notre kart. **items_position** garde tous les emplacements d'items, même ceux derrière notre kart.

items_type

Type : tuple
Tuple d'entiers des types d'items (**items_type**[i] correspond au type de l'item situé au bout de **items_position**[i]). L'entier donné indique le type d'item au sol :

Valeur en entier	**item au sol**
0	BONUS_BOX
1	BANANA
2	NITRO_BIG
3	NITRO_SMALL
4	BUBBLEGUM
6	EASTER_EGG

On utilisera cette variable pour déterminer si on doit se diriger vers l'item, ou bien si au contraire on doit s'en éloigner (dans le cas de la banane ou du bubblegum, par exemple).

karts_position

Type : tuple
Tuple de vecteurs 3D des karts ennemis en fonction de notre kart.

Les vecteurs sont dans le référentiel de notre kart.
Le premier élément du tuple correspond au kart le plus en avant (en dehors du nôtre).
Cette variable nous permettra d'éviter la collision entre karts, et de savoir quand activer certains items, dans le cas où on est assez proche d'un kart ennemi.

paths_distance

Type : tuple
Tuple de plusieurs listes avec deux valeurs, chaque liste représente un vecteur 2D composé de deux float.
La map de Supertuxkart est divisée en noeuds et en segments, la variable **paths_distance** donne :

- La distance du nœud d'indice i par rapport au début de la course
- La distance du nœud d'indice i+1 nous par rapport au début de la course.

Ce qui nous permet de déterminer la longueur du segment sur lequel on est (si on est sur un segment court, on va s'attendre à un virage serré et inversement).

paths_width

Type : tuple
Tuple de plusieurs listes avec une valeur (float).
Représente la largeur du segment d'indice i.
La valeur est toujours positive (sur Abysses antédiluviennes, elle est en moyenne autour de 10).
Cette variable peut nous servir dans le cas où on est sur une piste très serrée, pour déterminer si on peut se permettre de tourner brusquement ou non.

paths_start

Type : tuple
Tuple de plusieurs listes avec 3 valeurs (float), qui représentent un vecteur 3D.
Chaque vecteur correspond au début d'un segment du circuit.
Les vecteurs sont dans le référentiel de notre kart.

paths_end

Type : tuple

Tuple de plusieurs listes avec 3 valeurs (float), qui représentent un vecteur 3D.

Chaque vecteur correspond à la fin d'un segment du circuit.

Les vecteurs sont dans le référentiel de notre kart.