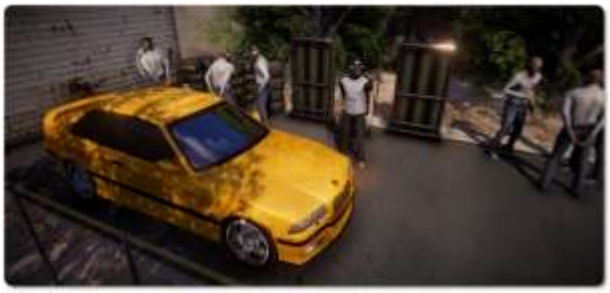
MIRAGESIM RACING TRAJECTOIRE LORS D'UN DRIFT MISSION 1

BIENVENUE !



Bienvenue dans l'écurie Mirage pour votre première mission **d'ingénieur d'essais** ! Vous allez prendre place à bord d'une voiture de sport afin de d'étudier le **comportement du véhicule lors d'une phase d'accélération**. Vous recevrez ensuite de la part de notre **ingénieur télémétrie**, les **données physiques** du véhicule au **cours du temps**.

COMMANDES UTILISEES

Accélération

Freinage

Changer la **Caméra**. Certaines vues permettent un pilotage plus précis.

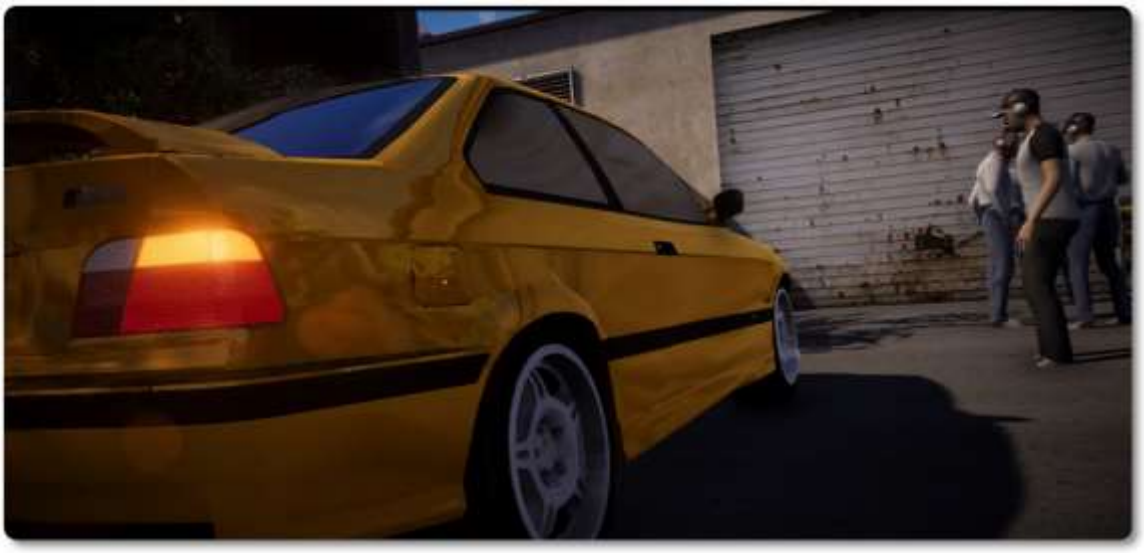
Ralenti. Mode qui vous donne le temps d'ajuster au mieux la trajectoire

Activer / désactiver l'**acquisition** des données pendant tout le mouvement

Tourner à gauche. Appuyer par petites touches pour ajuster la trajectoire

Tourner à droite. Appuyer par petites touches pour ajuster la trajectoire

TRAVAIL A REALISER



Sélectionner dans la simulation « Mission A1 ». Temps d'échantillonnage = 0,25 s.

I. Introduction

1. Trouver l'origine du repère (c'est-à-dire x=0 et y=0) en vous déplaçant et le placer sur la carte à droite. (Appuyer sur la touche A pour afficher vos coordonnées)

   Le point rouge indique votre position initiale >>>

2. Tracer les axes x et y de votre repère sur la carte à droite.

Conduite / Acquisition :

Réaliser l'acquisition d'un mouvement rectiligne accéléré du véhicule pendant quelques secondes dans la longueur de la piste.

A la fin de votre acquisition, utiliser la touche ***Copier*** pour copier les données de télémétrie et le raccourci CTRL+V pour coller vos données dans votre IDE Python préféré.

3. En regardant le script obtenu, et la variable t, indiquer le temps qui s'écoule entre deux mesures de positions ?

II. Calcul de vitesse sans boucle for

4. Le déplacement se fait selon l'axe y. Tapez à la suite du code issu de MirageSimRacing les instructions suivantes, à quoi servent-elles ?

```
print(y[0])
print(y[1])
print(y[2])
```

5. En utilisant la formule de la vitesse, modifier le programme obtenu afin d'afficher dans la console la vitesse entre le 1er et le 2ème point mesuré. (Les distances sont exprimées en m et le temps en s).

```
Vitesse0 =  y[                          # à vous de compléter
print(vitesse0)
```

6. Même question, mais pour les vitesses suivantes selon le même principe

```
Vitesse1 =
Vitesse2 =
Vitesse3 =
print(vitesse1)
print(vitesse2)
print(vitesse3)
```

7. Généralisation : Compléter la phrase suivante :

Pour calculer la vitesse10, je réalise la différence entre y[ ] et y[ ] divisé par le temps écoulé. On considère une valeur de i quelconque, pour calculer la vitessei, je réalise la différence entre y[ ] et y[ ] divisé par le temps écoulé.

III. Calcul de vitesse avec une boucle for

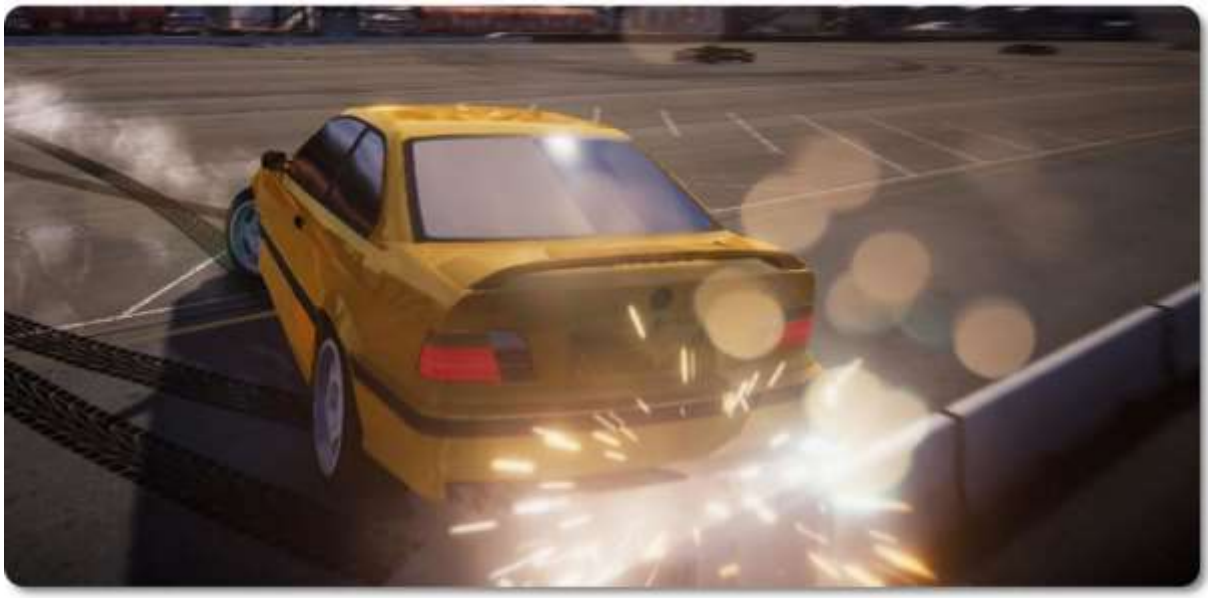
Comment fonctionne une boucle for ?
https://youtu.be/HzZ0TfoWOGM

8. Compléter le **code suivant** en l'intégrant **à la suite de votre programme** afin d'afficher dans la console **toutes** les vitesses à chaque point, en précisant l'unité.

```
for i in range (len(y)-1):
    vitesse = . . . . . # modifier cette ligne
    vy.append(vitesse) #ajout de la vitesse calculée à une liste
print(vy) #affichage de toutes les valeurs de vitesse
```

9. Quelle est la vitesse maximale atteinte en $km.h^{-1}$? ($10\ m.s^{-1} = 36\ km.h^{-1}$)

IV. Drift et vecteur vitesse



Conserver votre code précédent dans une fenêtre de votre éditeur Python, cela pourra vous resservir. Ouvrir à côté une nouvelle fenêtre vierge.

Sélectionner dans la simulation « Mission A2 Drift ». Temps d'échantillonnage = 0,25 s.

Conduite / Acquisition :

Réaliser en « drift », (en dosant votre accélération), un virage autour d'un plot. Votre temps d'acquisition doit être **inférieur à 5 secondes.**

Exemple de drift à réaliser

https://youtu.be/RK7GChGJxsg

A la fin de votre acquisition, utiliser la touche ***Copier*** pour copier les données de télémétrie et le raccourci CTRL+V pour coller vos données dans votre IDE Python préféré.

IV.a : Calcul des vitesses selon x et y :

10. Le mouvement se fait maintenant **sur les axes x et y**. Compléter le **code suivant** en l'intégrant **à la suite de votre programme** afin d'afficher dans la console **toutes** les vitesses à chaque point, en $m.s^{-1}$

```
for i in range (len(x)-1):
     vitesse = . . . . . .
     vx.append(vitesse)
print(vx)

for i in range (len(y)-1):
     vitesse = . . . . . .
     vy.append(vitesse)
print(vy)
```

Note : Les vitesses sont stockées dans les listes Vx et Vy, Nous allons les utiliser pour représenter des vecteurs vitesse.

IV.b : Représentation graphique du mouvement :

11. Utiliser le Document suivant pour réaliser le graphique des positions successives (x,y) du véhicule. *Représenter le graphique obtenu :*

DOCUMENT : TRACER UN GRAPHIQUE EN PYTHON AVEC MATPLOTLIB

Charger la librairie pour réaliser les graphiques (déjà importé)

```
import matplotlib.pyplot as plt
```

Ajout d'une grille en arrière-plan

```
plt.grid()
```

Création d'une échelle identique sur les deux axes (indispensable)

```
plt.gca().set_aspect('equal', adjustable='box')
```

Ajout des légendes indispensables au graphique

```
plt.xlabel('à compléter')
plt.ylabel('à compléter')
plt.title('à compléter')
```

Définition des axes et du style des points

```
plt.plot(x, y,'ro',label=' ')
```

Affichage du graphique

```
plt.show()
```

12. Comment peut-on nommer la trajectoire obtenue ?

13. La vitesse est-elle toujours la même sur toute la trajectoire ?

DOCUMENT : TRACER DES VECTEURS VITESSE EN PYTHON AVEC MATPLOTLIB

Configuration initiale du graphique (voir Doc précédent)

```
plt.grid()
plt.xlabel("à compléter")
plt.ylabel("à compléter")
plt.title("à compléter")
```

Ajout une seule flèche, dont la longueur est basée sur les valeurs de v_x et v_y

```
echelle = 0.1 #permet de réduire la taille des flèches
plt.arrow(x[1],y[1],echelle*vx[1],echelle*vy[1],head_width=1,
head_length=1)
```

Astuce : Cette ligne n'ajoute qu'un seul vecteur vitesse, mais vous pouvez modifier cette ligne et l'imbriquer dans une boucle for pour représenter tous les vecteurs vitesse.

Affichage du graphique

```
plt.plot(x,y,'ro')
plt.show()
```

Régler la longueur des flèches

Si les flèches sont trop longues ou trop petites, ajuster le coefficient `echelle`

Régler la taille de la tête des flèches

Si la tête des flèches ne s'affiche pas correctement, vous pouvez modifier le paramètre `head_length` et `head_width`.

IV.c : Représentation des vecteurs vitesse :

14. Utiliser le **Document suivant** pour ajouter sur le graphique les vecteurs vitesse pour chaque point.
    Représenter le graphique

15. Expliquer l'utilité des différentes parties de la ligne commençant par plt.arrow (….

16. Dans le code issu de MirageSim Racing, vous remarquez 2 séries de mesures sont réalisées, l'une pour l'arrière et l'autre pour l'avant du véhicule. Tracer sur un même graphique les trajectoires de l'avant et de l'arrière du véhicule avec les vecteurs vitesses. Commenter la courbe obtenue par rapport à la trajectoire « drift ».

PROGRAMME DE LA CLASSE DE SECONDE

Déroulement de la séance :

- L'activité se réalise en demi-groupe lors d'une séance de TP avec un IDE python de votre choix
- Les élèves ont déjà eu une introduction à Python dans le cours précédent
- L'activité se déroule en deux séances de 1h30. Libre à vous de découper autrement cette proposition d'activité pédagogique en fonction de vos contraintes, ou de rajouter des questions sur les thématiques non abordées du programme de seconde.

Programme :

Décrire un mouvement

- Choisir un référentiel pour décrire le mouvement d'un système.
- **Capacité numérique :** représenter les positions successives d'un système modélisé par un point lors d'une évolution unidimensionnelle ou bidimensionnelle à l'aide d'un langage de programmation.
- Définir le vecteur vitesse moyenne d'un point.
- Approcher le vecteur vitesse d'un point à l'aide du vecteur déplacement.
- Caractériser un mouvement rectiligne uniforme ou non uniforme.
- **Capacité numérique** : représenter des vecteurs vitesse d'un système modélisé par un point lors d'un mouvement à l'aide d'un langage de programmation.

Modéliser une action sur un système

- Représenter qualitativement la force modélisant l'action d'un support dans des cas simples relevant de la statique.

Principe d'inertie

- Exploiter le principe d'inertie ou sa contraposée pour en déduire des informations soit sur la nature du mouvement d'un système modélisé par un point matériel, soit sur les forces.