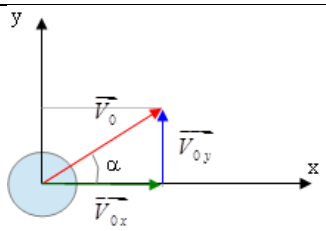
But du TP : Etudier expérimentalement un mouvement.

Le Blob Jumping est la nouvelle activité à la mode, qui débarque d'Outre-Atlantique. Cette nouvelle activité à sensation forte va en ravir plus d'un !

Des données ont été collectées à partir d'une vidéo du record du monde de Blob Jumping. A partir de ces données, vous allez déterminer les conditions initiales qui ont permis le record du monde.

Document 1 : Mouvement d'un objet lancé dans un champ uniforme

Un objet lancé avec une vitesse non-nulle dans un champ uniforme a une trajectoire parabolique.



L'équation de la trajectoire est alors :

$$y(x) = -\frac{1}{2}\frac{g}{v_0^2\cos^2\alpha}x^2 + x\tan\alpha + h$$

Flèche et portée d'une trajectoire parabolique sont liées à la vitesse initiale V_0 et à l'angle initial du saut α.

• Flèche du saut y_F : hauteur maximale atteinte par le centre d'inertie du système.

$$y_F = \frac{v_0^2\sin^2(\alpha)}{2g}$$

• Portée du saut x_P : distance entre le point de lancer et le point d'impact sur l'axe horizontal

$$x_P = \frac{v_0^2\sin(2\alpha)}{g}$$

Donnée : $g = 9{,}81\ m.s^{-2}$

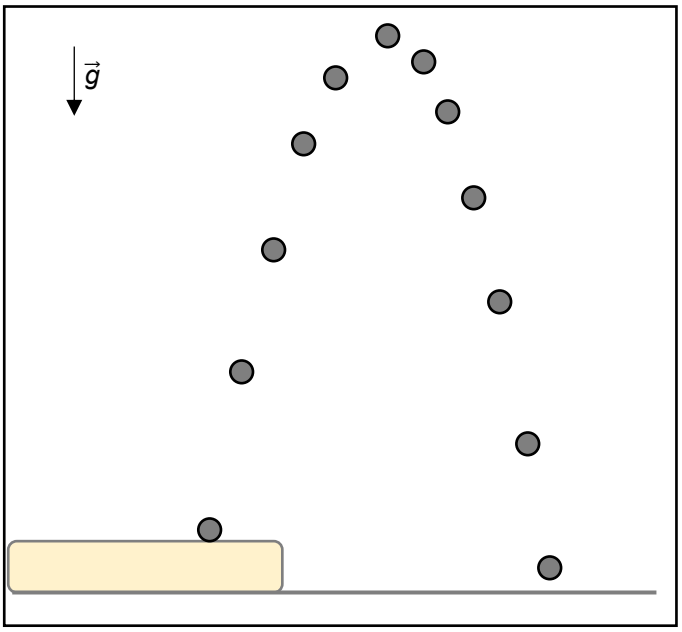
Document 2 : Le Blob Jumping

Le principe du Blob Jump est : u*n maximum de sensation, dans un minimum d'espace !*

L'activité tendance le « Blob Jumping » qui a déjà fait ses preuves sur les bases aquatiques d'Outre-Atlantique, débarque en France.
Le Blob Jump est un immense réservoir d'air posé dans l'eau. Le Blobber est celui qui est garantie d'avoir un maximum de sensation car il est situé à l'extrémité du Blob. Le « Jumper » est la personne qui va sauter d'une plate-forme situé à l'autre extrémité et qui va provoquer le catapultage du Blobber de plusieurs mètres dans les airs.

Le Blob Jumping est un mélange parfait de soleil, de fun et d'eau ! Il est le dernier né des sports à sensation, dans la lignée des Jetski, banane et parachute ascensionnel.

Chronographe d'un saut



Partie I : Etude préliminaire

Un schéma du chronographe du saut est présenté dans le document 2. On néglige les frottements de l'air.

1. Préciser le système et le référentiel d'étude.

2. Placer judicieusement le repère d'étude afin de faciliter les mesures de la flèche et de la portée (la distance horizontale maximale).

3. Représenter la trajectoire puis tracer la flèche du saut (la hauteur maximale). On peut considérer que le trampoline est à hauteur d'eau.

4. Tracer le vecteur vitesse initiale à l'instant t = 0 s, noté $\overrightarrow{v_0}$, sans soucis d'échelle, ainsi que ses composantes (projections) verticale $\overrightarrow{v_{0y}}$ et horizontale $\overrightarrow{v_{0x}}$. Faire aussi apparaître l'angle α.

5. Faire le bilan des forces appliquées sur le système avec leurs coordonnées (projections sur les axes) et dessiner le(s) vecteur(s) force sur l'un des points du chronographe.

Appeler le professeur pour vérifier les résultats avant de débuter le TP

Partie II : Etude de la trajectoire

Quelques instructions en python

- Un tableau est représenté avec des crochets [ ]. Les cases d'un tableau sont numérotées à partir de 0 (la première case est la case n°0). Pour un tableau nommé *L*, la valeur de la case n°*N* est représentée par *L[N]*.
- La fonction polyfit(abscisse, ordonnée, degré) permet de modéliser des données *(abscisse, ordonnée)* par un polynôme de degré donné. La fonction renvoie alors le tableau des coefficients du polynôme.

  Exemple : coef = polyfit(u, w, 2) renvoie les coefficients A=coef[0], B=coef[1] et C=coef[2] tels que $w = A\,u^2 + B\,u + C$

Rappels : Un polynôme de degré 0 est une constante ; degré 1 une fonction affine ; degré 2 une parabole

6. Proposer un protocole expérimental détaillé afin d'obtenir l'équation de la trajectoire à l'aide de LatisPro à partir de la vidéo du record. Préciser les grandeurs à obtenir lors de l'acquisition et quel graphique doit être obtenu lors de l'analyse.

Appeler le professeur pour vérifier et évaluer le protocole

Le protocole d'acquisition a été mis en œuvre et les données ont été exportées dans un fichier .csv nommé Donnees_BlobJump.csv, fournis par le professeur.
Le code python BlobJumping_Trajectoire.py permet de modéliser la trajectoire du sauteur à partir d'un fichier contenant les données de pointage, puis de tracer points et modèle sur un graphique.

7. Les lignes 20 et 21 du code sont incomplètes ; choisir parmi les propositions suivantes la bonne modélisation (fonction polyfit), en justifiant.

Proposition A

```
20 | coef = np.polyfit(t, x, 2)
21 | Trajectoire = [coef[0]*t_model**2+ coef[1]*t_model + coef[2] for t_model in t]
```

Proposition B

```
20 | coef = np.polyfit(t, y, 2)
21 | Trajectoire = [coef[0]*t_model **2+ coef[1]*t_model + coef[2] for t_model in t]
```

Proposition C

```
20 | coef = np.polyfit(x, y, 2)
21 | Trajectoire = [coef[0]*x_model **2+ coef[1]*x_model + coef[2] for x_model in x]
```

Proposition D

```
20 | coef = np.polyfit(x, y, 1)
21 | Trajectoire = [coef[0]*x_model + coef[1] for x_model in x]
```

8. A partir de la ligne 24, le code permet de tracer points et modèle sur un graphique. Les lignes 33, 34 et 35 du code sont incomplètes ; les compléter comme indiquer dans les commentaires du code.

Appeler le professeur pour valider la réponse. Compléter ensuite les lignes dans le code puis l'exécuter.

9. A l'aide des résultats de la modélisation de l'équation de la trajectoire, en déduire l'angle de tir α et la valeur de la vitesse initiale v_0.

10. Calculer la flèche du saut.

Appeler le professeur pour évaluer les résultats

Le code python BlobJumping_VitesseAcceleration.py permet de modéliser les positions, vitesses et accélérations du sauteur à partir d'un fichier contenant les données de pointage, puis de tracer points, modèles et vecteurs sur des graphiques.

11. Certaines lignes du code, correspondant aux différentes modélisations des positions, vitesses et accélérations, sont incomplètes : il manque l'ordre du polynôme associé au modèle dans la fonction polyfit. Compléter les fonctions polyfit ci-dessous avec l'ordre (0, 1 ou 2) du polynôme.

position sur x

```
81 | coef_x = np.polyfit(t, x, ......)
```

vitesse sur x

```
90 | coef_Vx = np.polyfit(t, vx, ......)
```

accélération sur x

```
99 | coef_Ax = np.polyfit(t, ax, ......)
```

position sur y

```
85 | coef_y = np.polyfit(t, y, ......)
```

vitesse sur y

```
94 | coef_Vy = np.polyfit(t, vy, ......)
```

accélération sur y

```
103 | coef_Ay = np.polyfit(t, ay, ......)
```

Appeler le professeur pour valider la réponse. Compléter ensuite les lignes dans le code puis l'exécuter.

Note : agrandir la fenêtre des graphiques pour une meilleure lisibilité

12. Commenter sur la direction et le sens des vecteurs vitesse et accélération.

13. A partir des modèles obtenus, commenter et <u>justifier</u> avec les lois de Newton l'évolution de v_x et de v_y en fonction du temps.

14. Déterminer $v_x(0)$ et $v_y(0)$ à partir des modèles obtenus puis calculer par trigonométrie, à l'aide du document 1, l'angle de tir α. Cette valeur est-elle en accord avec celle trouvée précédemment ?

15. En vous aidant du document 1, indiquer sur quels paramètres les concurrents peuvent agir pour espérer augmenter leur record.