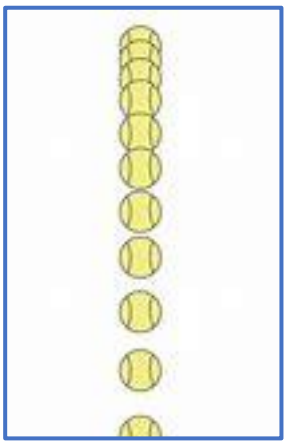
Unité 5 : Utiliser la librairie TI System & Ti_PlotLib | **Application : Etude de la chute libre**

Dans cette application de l'unité 5, vous allez réinvestir les notions vues lors dans les unités 4 et 5 afin de créer un simulateur permettant de décrire un mouvement.

Objectifs :

- Effectuer une simulation d'une chronophotographie.
- Exporter les résultats dans les listes de la calculatrice.



On propose dans cette application de l'unité 5, d'utiliser le langage Python afin de créer un simulateur d'un phénomène physique. Nous retenons l'étude de la chute libre :

- Le script devra permettre la description d'un mouvement ;
- La représentation graphique des données sous forme d'une chronophotographie ;
- L'export des données vers les listes de la calculatrice.

Le problème : Une balle est lâchée sans vitesse initiale d'une hauteur h. Les photographies sont effectuées au cours du temps toutes les 60 ms.

Vous allez écrire un script permettant de calculer, au cours du temps, la position de la balle, puis de la représenter graphiquement.

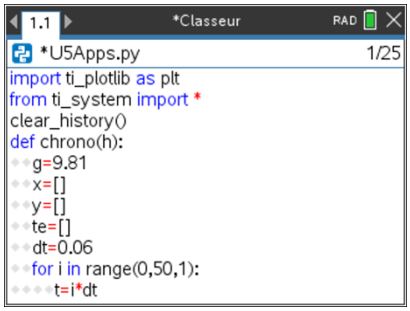
a) Calcul des positions successives

Lors de la chute libre, à partir d'une origine des temps et des espaces choisis (y>0), la position de la balle peut être calculée en utilisant la relation $y = -\frac{g}{2} \times t^2 + h_0$.

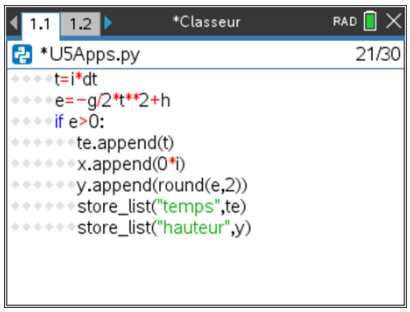
Rappelons que : $g \approx 9{,}81$ m.s^{-2}.

Créer un nouveau script et le nommer U5Apps.

- Importer les modules **TI System** et **TI PlotLib.**
- Nettoyer l'écran.
- Créer une fonction **chrono(h)** prenant comme paramètre la hauteur initiale à laquelle est lâchée la balle et qui affiche la position de la balle en fonction du temps.
- Créer trois listes vides, abscisse **x**, ordonnée **y** et temps **te.**
- Les ordonnées sont calculées toutes les 60 ms.
- Les valeurs sont sauvegardées dans les listes si y >0 (la balle ne s'enfonce pas dans le sol).



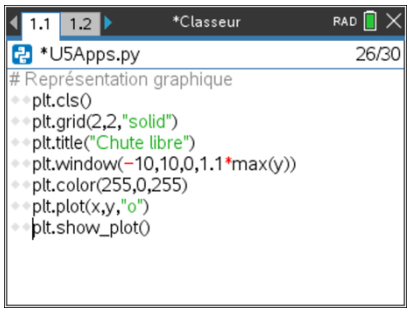
Créer une boucle fermée permettant de calculer l'altitude de la balle en fonction du temps. Les résultats sont exprimés à 10^{-2} près et stockés dans leurs listes respectives. Comme on souhaite avoir une représentation graphique de la chronophotographie correspondant à un cas réel, la liste des abscisses est complétée avec des 0.
Enfin, les listes **te** et **y** sont respectivement exportées dans les listes **temps** et **hauteur** de la calculatrice.



b) Représentation graphique

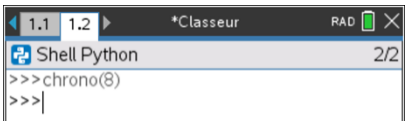
Paramétrer une représentation graphique :

- Nettoyer l'écran **plt.cls( )**.
- Afficher une grille d'unité 2 **plt.grid(xscl, yscl, type,(r,v,b))**.
- Régler la fenêtre graphique **plt.window(x_{min}, x_{max}, y_{min}, y_{max})**.
- Fixer la couleur du point au magenta **plt.color(255,0,255)**.
- Représenter le nuage de point **plt.plot(x-list, y-list, marque)**.
- Afficher le graphe **plt.show_plot( )**.



Exécuter votre script et appeler la fonction chrono [var], puis lui fournir comme argument la hauteur initiale du lâché (8m par exemple).

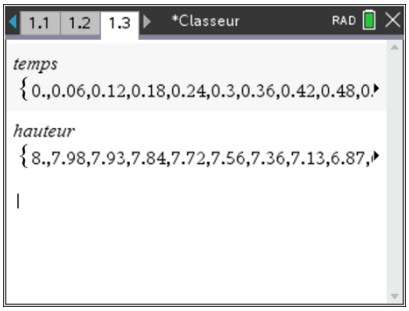
La chronophotographie est représentée.



Conseil à l'enseignant : A partir de la liste des positions de la balle, on pourra éventuellement calculer la vitesse de la balle, puis afficher les vecteurs vitesses.

c) Visualisation des données exportées

- Quitter l'éditeur Python et afficher les listes.
- Insérer une application **Calculs**.
- Rappeler les listes **temps** et **hauteur**.



- Insérer une application **Graphiques** et représenter le nuage de points (**temps**, **hauteur**).
- Utiliser l'outil **Trace** afin d'explorer la représentation graphique.