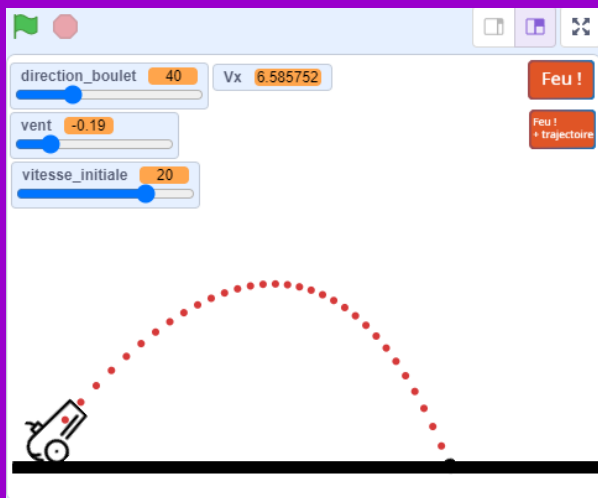




# 6 Balistique : Tir au canon



À quoi ressemble la trajectoire d'un boulet de canon et comment peut-on la simuler avec Scratch ?  
Voyons cela ...



▾ Vidéo

<https://jeunesingenieurs.be/vid-s4-3>



▾ Proposition de solution



<https://scratch.mit.edu/projects/1083329208>



# En mode défi

## ▼ Tu te lances en mode Défi ?

Voici une proposition de méthode.

1. Prépare des objets (sprites) : le canon, le boulet, le sol et un bouton pour déclencher le tir. Ajoute un petit effet au bouton pour qu'il donne l'impression de s'enfoncer quand on appuie dessus.
2. Programme le tir du boulet en trajectoire rectiligne, en fonction de la direction du canon. La trajectoire rectiligne est celle qui serait observée en absence de gravité (attraction terrestre).
3. Modifie ton code afin de prendre en compte la gravité qui va ralentir le mouvement ascendant et au contraire, l'accélérer dans la phase descendante.
4. Duplique ton code et le bouton déclencheur et modifie-le de façon à pouvoir tracer et observer plus facilement la trajectoire .
5. Programme l'effet du vent qui peut être favorable ou contraire.



## 6.1 LES SPRITE

S

### 🎯 Objectif

Prépare ton  
projet :



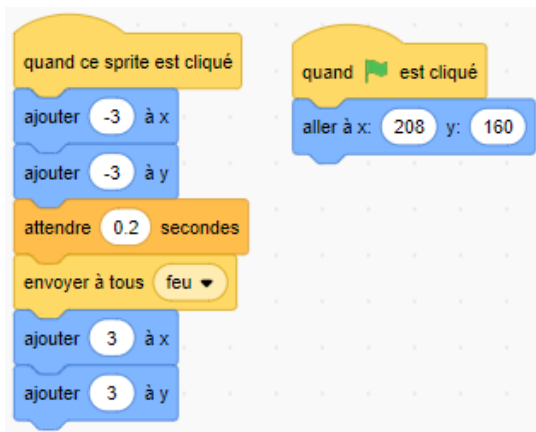
### ▀ Les sprites.

- Le boulet est un simple disque.
- Le canon peut être dessiné ou trouvé ici :  
<https://www.svgrepo.com/vectors/cannon/>
- Le sol : une ligne épaisse.

- Un bouton pour déclencher le tir.

## ■ Ajoute ce code au sprite **Bouton 1**

Ce code très simple déclenche le tir quand on clique sur le bouton avec en plus un petit effet « bouton enfoncé » pour augmenter le réalisme.





## 6.2 TIR AU



CANO  
N  
EN  
L'ABSE  
NCE

## 🎯 Objectif

Programmer un tir au canon en l'absence de gravité en tenant compte d'un angle de tir.

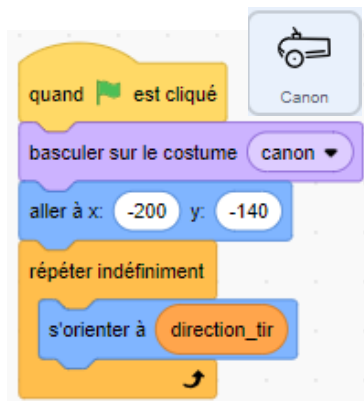
En l'absence de gravité, le projectile partirait en ligne droite.

➤ Ajoute ce code au **Boulet**

➤ Crée deux variables : *direction\_tir* et *vitesse\_initiale*.

➤ Ajoute ce code au **Canon**

Avec ce script, le canon va s'orienter comme le boulet. En réalité, c'est le contraire qui se produit mais au niveau du visuel, cela ne change rien.





# TIR AU CANON EN L'ABSENCE DE GRAVITÉ

## ► Affiche les variables en mode *Barre de défilement*

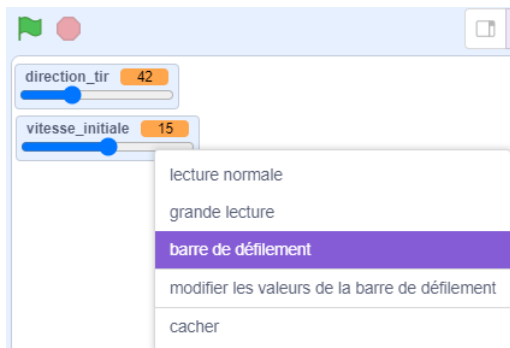
Affiche ces variables en mode *barre de défilement* et affecte leur ces valeurs limites :

***direction\_tir :***

entre 20 et 90°

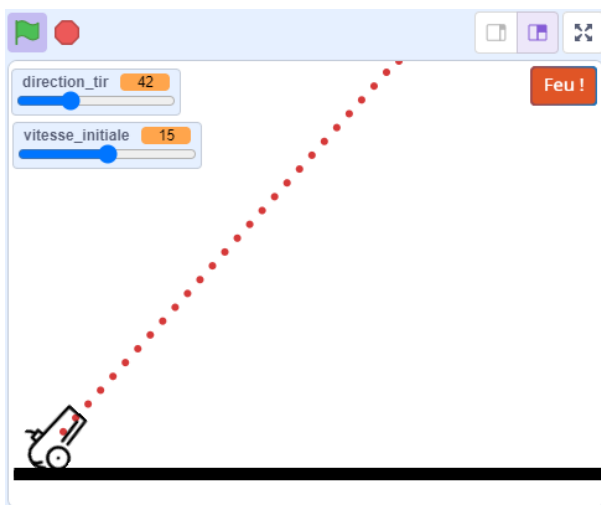
***vitesse\_initiale :***

entre 5 et 25pas



## ► Teste ton code

Le boulet devrait se déplacer comme ceci si tu choisis la direction 42°.





## 6.3 ON AJOUT

E LA  
GRAVI  
TÉ :  
LE TIR  
BALSIT  
IQUE

Et c'est ici que les choses se compliquent...un peu.

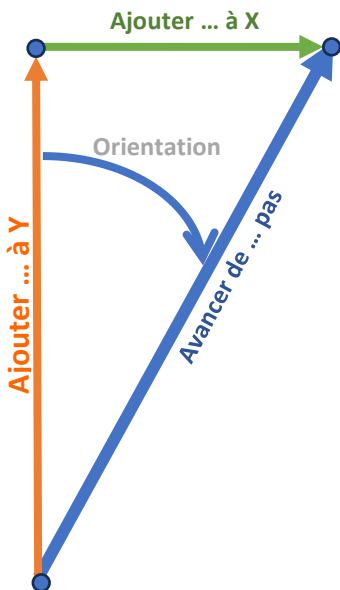


## 🎯 Objectif : prendre en compte la gravité

Nous avons simulé la trajectoire du boulet à l'aide des blocs **s'orienter à** et **avancer de ... pas**.

(voir la carte 3-4)

Or, la gravité agit sur la composante verticale du déplacement. Il faudrait donc remplacer le bloc **avancer de ...pas** par deux blocs : **ajouter ... à x** et **ajouter ... à y** pour pouvoir ensuite simuler l'effet de l'attraction terrestre. Il faudra modifier la vitesse de déplacement vertical pendant que la vitesse horizontale elle, reste constante.



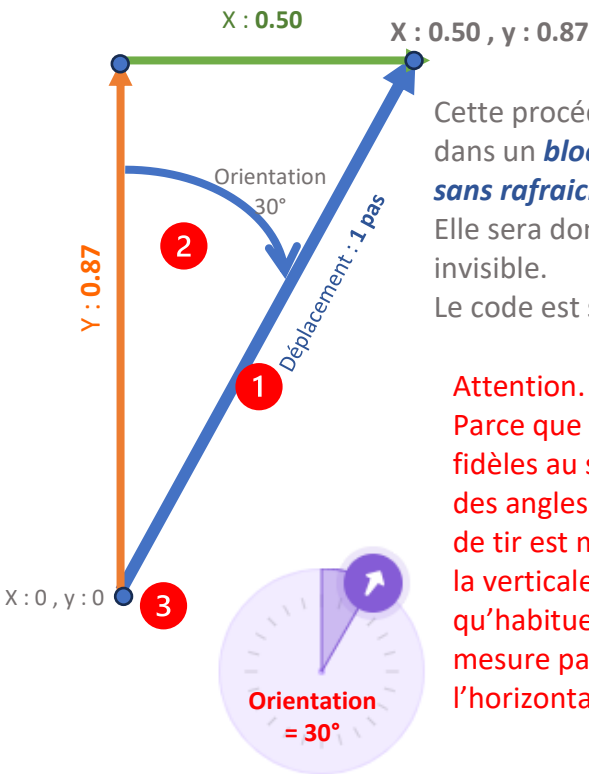
Nous proposons deux méthodes pour réaliser ceci : à l'aide d'un test ou avec la trigonométrie. Si tu n'as jamais fait de trigonométrie, contente-toi de la première...



# ON AJOUTE LA GRAVITÉ : LE TIR BALSITIQUE

## Transformer la vitesse initiale en $V_x$ et $V_y$ .

Avant d'effectuer le tir, nous simulons ❶ un déplacement de 1 pas en prenant en compte de ❷ l'angle de tir et ce, à partir du ❸ point ( $x : 0$ ,  $y : 0$ ) de la scène.



Cette procédure sera réalisée dans un **bloc personnalisé** et **sans rafraîchissement** d'écran. Elle sera donc instantanée et invisible. Le code est sur la carte suivante.

### Attention.

Parce que nous devons rester fidèles au système de mesure des angles de Scratch, l'angle de tir est mesuré par rapport à la verticale alors qu'habituellement, on le mesure par rapport à l'horizontale.



# ON AJOUTE LA GRAVITÉ : LE TIR BALSITIQUE

## Le code

Testé avec un angle de tir de 30°, les valeurs enregistrées de l'ordonnée y et de l'abscisse x sont 0.8660 et 0.5000.

The code is organized into two main sections. The first section, triggered by a 'when green flag is clicked' event, sets initial values: 'vitesse\_initiale' to 15 and 'gravité' to -1. It then triggers a 'when I receive feu' event, which sets the direction to 'direction\_tir', moves the sprite to (-206, -130), and displays a message. The second section is a loop that repeats until the sprite 'touches le Sol'. Inside the loop, it adds 'Vy' to 'y', adds 'Vx' to 'x', and adds 'gravité' to 'Vy'. A separate block titled 'définir Calcul Vx et Vy par TEST' shows the calculation of velocity components: 'Vy' is set to 'vitesse\_initiale \* ordonnée y', and 'Vx' is set to 'vitesse\_initiale \* abscisse x'. Red arrows point from these calculation blocks to the final output values for 'ordonnée y' and 'abscisse x'.

```
quand [drapeau vert] est cliqué
  mettre vitesse_initiale à 15
  mettre gravité à -1

quand je reçois feu
  s'orienter à direction_tir
  Calcul Vx et Vy par TEST
  aller à x: -206 y: -130
  montrer

  répéter jusqu'à ce que touche le Sol
    ajouter Vy à y
    ajouter Vx à x
    ajouter gravité à Vy
  cacher
```

Testé avec un angle de tir de 30°, les valeurs enregistrées de l'ordonnée y et de l'abscisse x sont 0.8660 et 0.5000.

ordonnée y: 0.8660254037844386

abscisse x: 0.5000000000000001



# ON AJOUTE LA GRAVITÉ : LE TIR BALSITIQUE

Une alternative : avec la trigonométrie

À ignorer si tu n'as pas fait de trigonométrie !

Soit, un angle de tir de  $30^\circ$  par rapport à la verticale.

The Scratch script is as follows:

- when green flag is clicked**
  - set `vitesse_initiale` to 15
  - set `gravité` to -1
- when I receive feu**
  - turn to `direction_tir`
  - call `Calcul Vx et Vy TRIGO` block
  - go to x: -206 y: -130
  - show
  - repeat until** touches `Sol`?
    - add `Vy` to `y`
    - add `Vx` to `x`
    - add `gravité` to `Vy`
  - hide
- define block: Calcul Vx et Vy TRIGO**
  - set `Vx` to `vitesse_initiale` \* `sin` of `direction_tir`
  - set `Vy` to `vitesse_initiale` \* `cos` of `direction_tir`

Trigonometric values shown:

- `cos` of `direction_tir` = 0.8660254038
- `sin` of `direction_tir` = 0.5

Besoin d'explications ?

J.P. B

Appelle ton prof de Maths ou de Physique 😊

Série 3

SCRATCH  
CARTES

6-12



## 6.4 OBSE RVON

S LA  
TRAJE  
CTOIR  
E  
DE  
PLUS  
PRÈS

### 🎯 Objectif

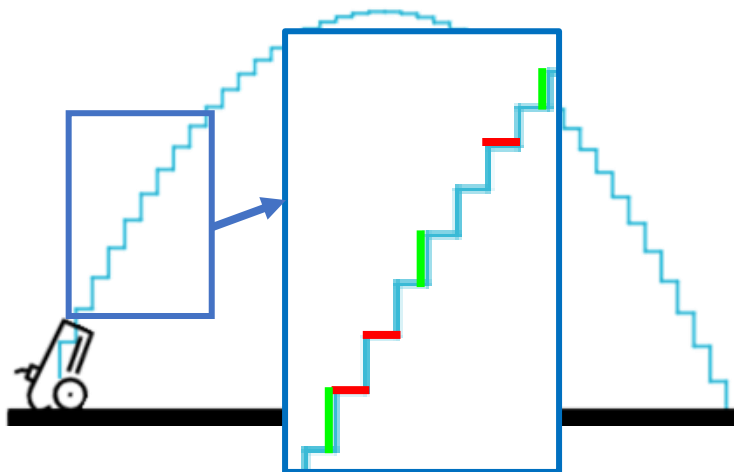
Avec les blocs de l'extension **styo**, tracer la trajectoire afin d'observer les caractéristiques du tir balistique.

Le code : <https://scratch.mit.edu/projects/138565118>

### ▼ Trace la trajectoire

Les « escaliers » ne sont pas visibles lors du déplacement du boulet car les deux blocs **ajouter  $V_y$  à  $y$**  et **ajouter  $V_x$  à  $x$**  s'exécutent de manière simultanée.

Dans la partie ascendante,  **$V_y$**  décroît (= décélération). Dans la partie descendante, le déplacement vertical est accéléré.  **$V_x$**  par contre reste constante.





## 6.5 ÇA DÉPEN

D...

SI Y'A

DU

VENT

...

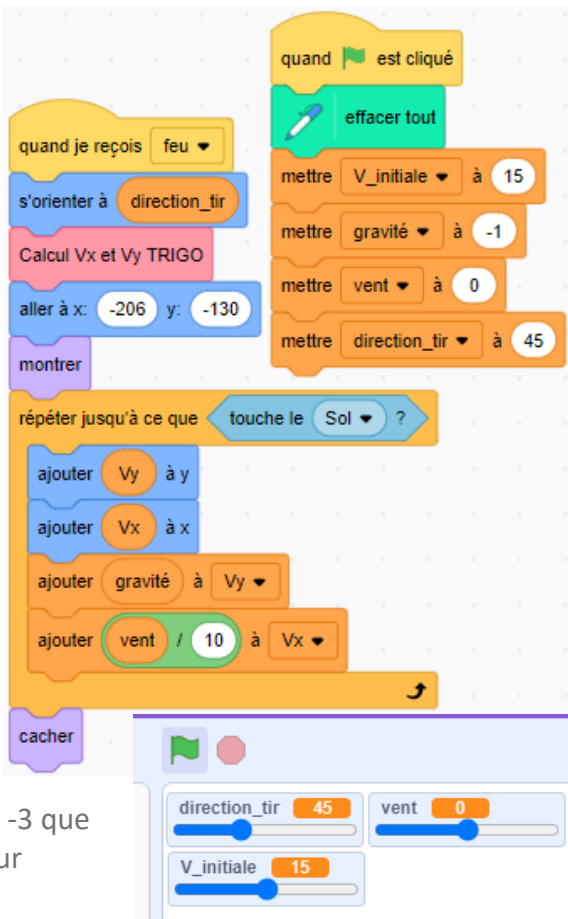
## 🎯 Objectif : ajouter le facteur vent

Un vent de face réduit la trajectoire et un vent de dos l'allonge. Nous additionnons une valeur à

**V<sub>x</sub>** comprise entre -0.3 pour un vent de face et + 0.3 pour un vent de dos. Le choix de ces valeurs est empirique.

### ► Modifie ces deux scripts

La variable vent prend les valeurs comprises entre 3 et -3 que l'on divise par 10 pour obtenir 0.3 à -0.3.







## 6.6 À TOI DE

JOUER



### ■ Quelques propositions d'améliorations.

- Modifie le code de façon à ce que l'angle de tir soit mesuré à partir de l'horizontale. Adapte tes calculs si nécessaire.
- Ajoute un objet à atteindre et un obstacle à surmonter. Tu pourrais créer un jeu qui s'inspire de Angry Birds...
- Ajoute une réserve de boulets et un score qui s'incrémente chaque fois qu'une cible est touchée.
- Ajoute un bouton qui permet de choisir un tir avec impression du tracé.