

Simulación computacional:

Tarea 2 - Movimiento en 2 dimensiones

Actividad:

1. Introducir, en el archivo sobre el que vienen trabajando, variables de forma tal de poder describir un movimiento bidimensional. Tendrán que incorporar en *Inicialización de variables globales > Variables cinemáticas y sus valores iniciales* las variables que consideren necesarias.

Recomendación: vayan guardando copias de las distintas versiones del archivo que van modificando, así, si en algún momento quieren volver atrás o necesitan simular un movimiento en una dimensión pueden trabajar sobre ellas.

2. En el bloque bajo el nombre de “tick()”, incluyan las líneas de código necesarias para actualizar los valores de dichas variables. No olviden incluir al comienzo de ese bloque una línea que diga “**global** nombre de la variable” para aquellas variables globales cuyo valor se modifique.
3. En el lugar donde se grafica el círculo que representa al objeto, modificar lo necesario para que represente correctamente la posición vertical.

```
# Aquí tenemos lo que se grafica
def draw(canvas):

    # De esta forma uno puede introducir texto en la representación
    # gráfica, dando la posición, tamaño y color de dicho texto.
    canvas.draw_text("Este movimiento es un MRU", (20, 625), 20, "White")
    # Dibuja un segmento que une los dos puntos especificados,
    # de ancho tres y de color gris
    canvas.draw_line((0,350),(WIDTH,350),3,"Grey")

    # Dibuja un círculo en la posición en la ventana indicada por
    # las coordenadas, en este caso "posx" , 350.
    # De color azul, radio= valor de la variable "radio" y
    # un borde de tamaño 2 y también de color azul
    canvas.draw_circle((posx,350),radio, 2,"Blue","Blue")
```

4. Simulen un MRU “en diagonal” (con velocidad con componentes horizontal y vertical no nulas). Analicen cuál es el sistema de referencia (dónde está el 0 de cada dirección y hacia dónde aumenta el valor de posición).
5. En el bloque de código “click2()”, hagan las modificaciones necesarias para que al clicar el botón **Reset**, el sistema vuelva a sus condiciones iniciales.
6. Simulen un “tiro oblicuo”. Vean cómo se modifica la trayectoria al cambiar el valor de la aceleración.

7. Si uno quisiese trabajar con posiciones medidas en unidades de longitud físicas y no en pixels (es decir que los valores que les damos a nuestras variables de posición, velocidad y aceleración estén en, por ejemplo, metros, metros sobre segundo, etc.), ¿cómo debería uno transformar esos valores a la hora de graficar la posición del objeto?

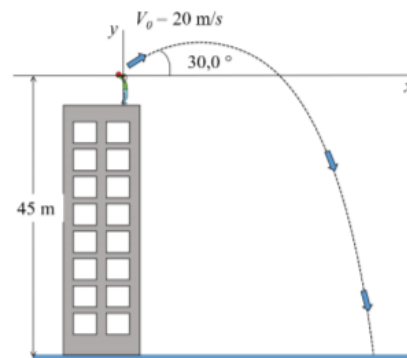
Vamos a procurar hacer una transformación que tenga las siguientes características:

- que sea una función lineal (en realidad, tendremos una función lineal diferente para cada coordenada).
- que podamos elegir el rango de valores que se representarán de alguna de las dos coordenadas x o y . Esto fijará una escala.
- que tenga la misma escala en x que en y (para que no altere la forma de las trayectorias).
- que podamos elegir cuál es el valor mínimo (o máximo) que queremos que se vea representado de la variable para la cual **no** indicamos el rango de valores a representarse.
- que podamos elegir si deseamos tomar los positivos hacia la izquierda o hacia la derecha para la coordenada x y que podamos elegir si deseamos tomar los positivos hacia arriba o hacia abajo para la coordenada y .

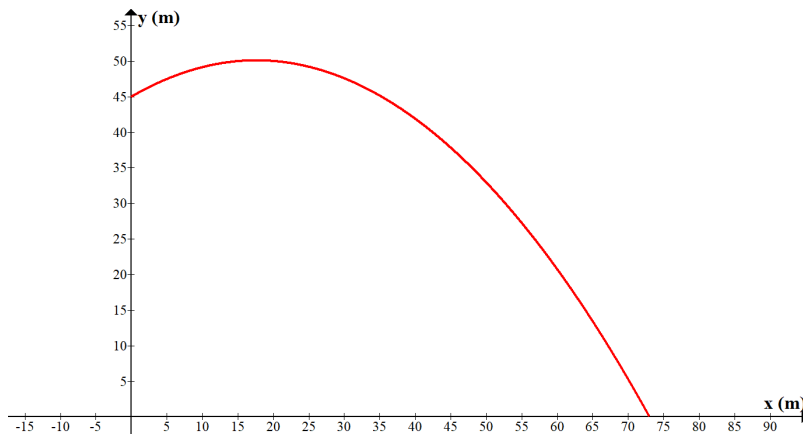
A modo de ejemplo, consideren la situación planteada en el siguiente problema:

Se lanza una piedra hacia arriba desde la parte superior de un edificio en un ángulo de $30,0^\circ$ con la horizontal y con una velocidad inicial de $20,0$ m/s. El punto de liberación está a $45,0$ m respecto de la superficie de la Tierra. Considere despreciable la resistencia del aire.

- a) ¿Cuánto tiempo le toma a la piedra golpear la superficie de la Tierra?
b) Determine la velocidad de la piedra en el impacto.
c) Encuentre el alcance horizontal de la piedra.



Esto da como resultado una trayectoria que es:



Supongamos que entonces queremos:

- que los valores de la variable de posición dentro del código estén expresados en metros.
- que la ventana abarque un rango de posiciones en y que vaya de -5m hasta 75m (la altura de la ventana en pixels está dada por el valor de la variable HEIGHT que aparece en la línea 14 del programa).
- que tenga la misma escala en x que en y .
- que el mínimo valor de x que se represente sea -5m.
- que el eje x esté orientado de forma que los positivos sean hacia la derecha.
- que el eje y esté orientado de forma que los positivos sean hacia arriba.

Algunas “ayudas”:

- queremos que sean funciones lineales a las cuales les damos posiciones en metros (o alguna otra unidad física) y nos dan como resultado posiciones en pixels dentro de la ventana.
- como se trata de funciones lineales, saber el resultado en pixels para dos valores de posición nos define la función.
- el valor absoluto de la pendiente nos dará la escala (cantidad de pixels/cantidad de metros que representan)
- cambiar el signo de la pendiente invierte el sentido de positividad de esa coordenada.
- algo que puede ser útil, es pensar cuál es el valor que uno quiere representar en la posición 0 de la pantalla para cada eje (ese valor sería la raíz de la función lineal y, con esa información y la pendiente, uno conoce completamente la función).

Para implementar esta transformación en el código, uno simplemente deberá modificar la línea en la cual se dibuja la partícula (mencionada en el punto 3 de esta tarea), poniendo, como las coordenadas donde se grafica, las expresiones de las funciones lineales construidas.