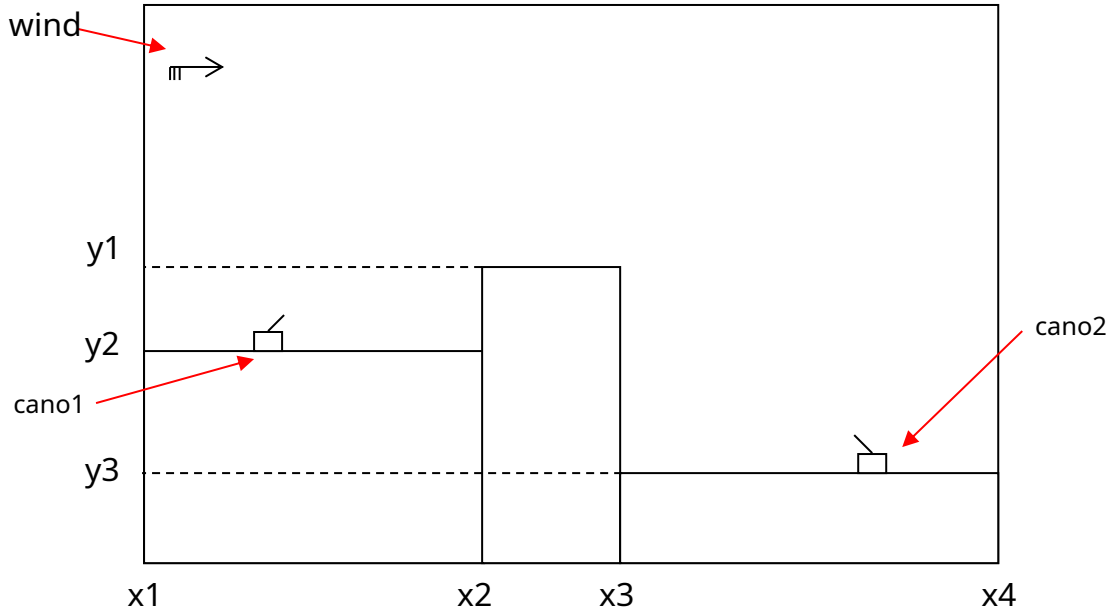
It is requested that a game be programmed to simulate the ballistic trajectory of projectiles fired by two cannons fired at each other, following the directions detailed below. The guns must be placed randomly within a scenario also generated randomly according to the following format:



on:

- the x1-x2 distance is a minimum of 100 pixels and a maximum of 400.
- the x2-x3 distance has a minimum of 20 pixels and a maximum of 150 pixels.
- y2 varies between 0 and 300.
- y1 varies between y2 and 300.
- y3 varies between 0 and y1.
- the cannons are 20x10 pixel rectangles with a 15 pixel line coming out of the top and are placed randomly respectively between x1-x2 and between x3-x4, without leaving the screen.
- at the top left there is a wind indicator (randomly generated direction and force), the direction can be right or left, the force can be between -5 and 5, if the value of the wind force is zero, then the arrow should not be drawn.

Scenario data (x values$_i$, of the y$_i$, wind force ...) will be saved in the list of properties of an atom that you will call "scenario".
The data of the cannons (angle, position and others that you may need) will be saved in the property lists of two atoms that you will call "cano" and "cano2".

In addition to those needed to draw this scenario, the functions must also be written:

- **(simulates cannon speed)**
  This function, given a cannon and the initial rate of fire, calculates and draws the trajectory of the projectile. The simulation ends if the projectile hits the ground, the middle mount, the other cannon, or if it exits on the side of the target cannon.
- **(raise cannon degrees)**
- **(low cannon degrees)**
  These two functions serve to control the initial angle of the shot, they must allow to raise or to lower the corresponding gun the number of degrees that is asked for.

- **(paints)**
  This function is responsible for initializing the stage, painting all the elements and placing the writing cursor at the top left of the screen, where all the instructions of the players will be entered.
  Alert! You must ensure that the instructions always enter this position (hint: use repeat)

Remember!
The equations of projectile shooting were studied and deduced by Galileo Galilei. The trajectory described by a projectile is a parabola that can be studied as a result of two motion, a uniform (constant velocity) along the axis***x***, and a uniformly accelerated (constant acceleration g = -9.8) along the axis ***y***.
The equations of this motion are as follows:

position

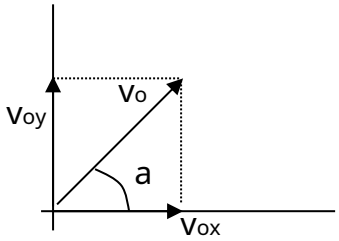
$x = v_x \cdot T$

$y = v_y \cdot T + \frac{1}{2} a \cdot t^2$

speed

$v_x = v_{ox}$

$v_y = v_{oy} + a \cdot t$



$v_{ox} = v_o \cdot \text{Cos } a$

$v_{oy} = v_o \cdot \text{Sin } a$

To paint the trajectory, the position of the projectile (x, y) must be continuously calculated from the position it is in and the updated speed.
We will consider the influence of wind on the trajectory of the projectile. So if there is wind, the speed on the axle***x*** it is also affected by the intensity of the wind (from -5 to 5) as if it were an acceleration (you can consider that the position is calculated the same as without acceleration, therefore, the only formula that changes is: $v_x = v_{ox} + a \cdot t$).

Trigonometric and rounding functions that may be useful to you:

(without x) :	calculates the sine of the angle x (expressed in radians).
(cos x):	calculates the cosine of the angle x (expressed in radians). this
(realpart (round x)):	combination converts the real number x to an integer. generates
(random x):	a random number between 0 and x.

Drawing functions

They act on the LISP screen, the coordinate (0,0) corresponds to the lower left point of the window. The dimensions of the drawing window are 640x340 pixels.

Alert! the drawing functions (move, moverel, draw, and drawrel) receive integers as parameters. That's why you have the rounding features we've proposed to you.

(cls) : completely clears the screen.
(cleol): delete all characters to the end of the line.
(goto-xy mn): Place the write cursor in column m and row n.

(color Re Ge Be Rf Gf Bf) :
sets the write color (Re Ge Be) and the background color (Rf Gf Bf), where R, G and B are the components of each color (from 0 to 255).

(move xy) : moves the point where the pencil is in position (x, y).

(moverel xy) : moves the point where the pencil is located x pixels horizontally and y pixels vertically unpainted.

(draw xy): move the pencil from where it is to the (x, y) position by painting the entire path.

(drawrel xy) : move the pencil x pixels horizontally and y vertically, painting the whole route.

Notes:

- This exercise should be done in groups of at most two people.
- The set and setq functions or any other function or structure not seen in class cannot be used.
- The access and modification functions can be used in the property lists of the stage, cano and cano2 atoms.
- Maximum delivery date: 25/04/2021.
- Delivery mechanism: upload a "canons.lsp" file to the task you will have in aulaDigital. Follow these rules:
  - At the beginning, and commented on, the file should include the full names of the two components of the group.
  - The file must contain the commented functions.
- The procedure to check the operation of the exercise will consist of reading the file with the command "> (load 'canons)" to later check the execution of the functions. Please check that this procedure works with your file before submitting it.