**Programming monitors to calculate the number of particles vs. position and time**

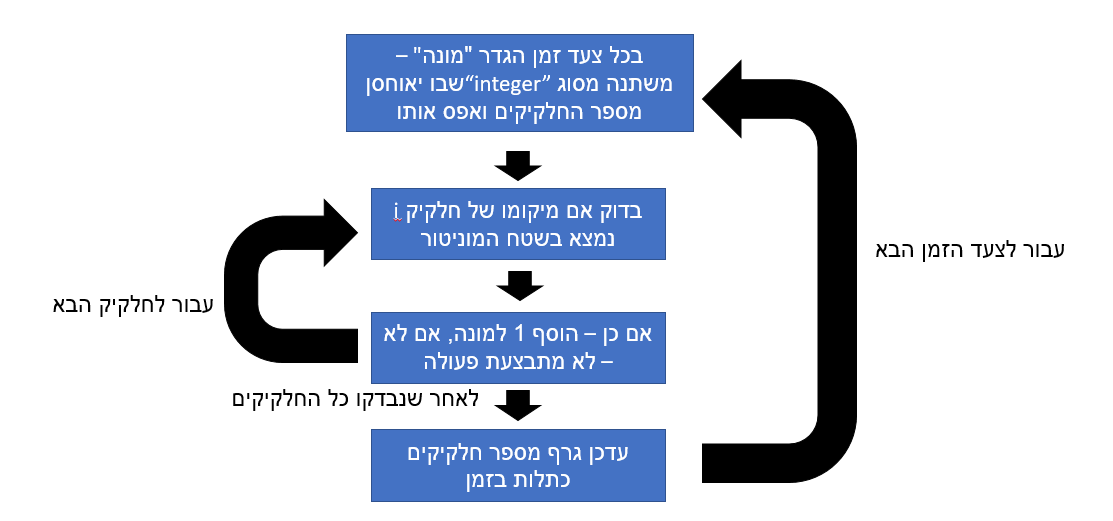
Download the simulation file (start\_here.py) from the course website.

Below are instructions that will help you build monitors to measure the number of particles vs. position and time.

Complete the lines and add them to the appropriate places in the code.

***Part A: Representing the number of particles vs. time***

We’ll build a monitor that calculates the number of particles in the center of the table vs. time, using the following algorithm:



1. **Making a graph**

Location of the graph window, and axis labels

We’ll display a graph by adding the following lines:

**gdisplay(x=350, y=0, width=350, height=350, xtitle='t', ytitle='N')**

**density\_t = gcurve(color=color.yellow)**

Name of the graph to which we will turn next (arbitrary choice)

1. **Representing the monitor with a rectangle**

We’ll represent the monitor with a rectangular box located at the center of the table.

**monitor = box(pos=(\_\_,\_\_,\_\_), size = (\_\_,\_\_,\_\_), color=color.yellow, opacity = 0.2(**

מידות המוניטור

מיקום המוניטור

1. **Defining and initializing a variable**

We’ll define a variable of type “integer” and set it to 0 in each time step.

**counter = 0**

1. **Counting the particles in the area of the monitor**

We’ll count how many disks are in the area of the monitor. We’ll add a condition that, when it is fulfilled, the number 1 is added to the variable “counter”.

**if disks[i].pos.x > \_\_\_\_\_\_\_\_ and disks[i].pos.x < \_\_\_\_\_\_: counter = \_\_\_\_\_\_**

1. **Updating values on the graph**

We’ll display a graph of the number of particles vs. time.

**density\_t.plot(pos = (\_\_\_\_\_,\_\_\_\_\_))**

***Part B: Representing the number of particles vs. position***

We’ll measure the distribution of the number of disks along the x-axis, and represent it with a histogram using the following algorithm:



1. **Producing a histogram**

We can produce a histogram graph in VPython using the following two lines:

Location of the graph window and the axis labels

**gdisplay(x=0, y=450, width=430, height=450, xtitle='x', ytitle='n')**

**density\_x = ghistogram(bins=arange(\_\_\_\_,\_\_\_\_,\_\_\_\_), color=color.\_\_\_\_)**

Name of the graph that we will turn to (arbitrary choice)

Maximum value on the x-axis

Column width

Minimum value on the x-axis

1. **Preparing a list with zeroes**

We’ll prepare a variable of type “list” – an **empty** list in which we will store (in the next step) the positions of the particles.

**positions = \_\_\_**

1. **Preparing and updating the list of positions of disks along the x-axis**

In the code that you have, there is already a list of positions of the balls: disks[i].pos.x . Add the positions of each of the particles to the list that we created in the previous stage.

**positions.append(\_\_\_\_)**

1. **Updating the histogram list**

To update the histogram, add the following line:

The list we want to display on the histogram graph

**density\_x.plot(data = \_\_\_\_\_)**