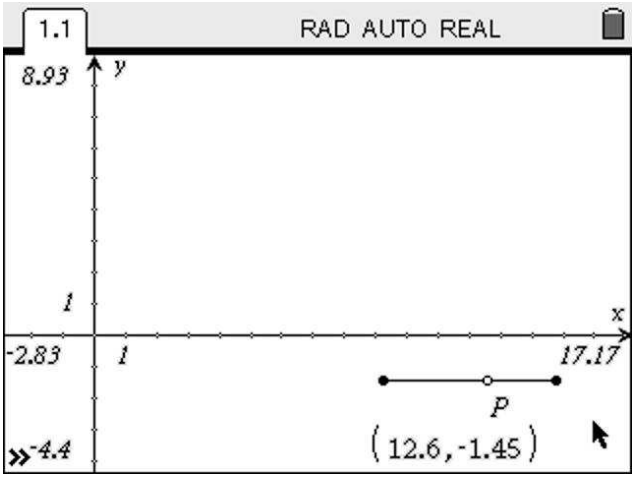
A simple dice roll simulator

This document describes how to build a 3 dice roll simulator, but only the generation and storage of events part, not the graphical display of actual dice with the proper dots marked. The most important thing is to be able to replicate the key aspects of any simulator and to be able to apply them to other constructions.

The first step to build any dynamic simulator is choosing a suitable control point. Usually a point over a segment does the job.

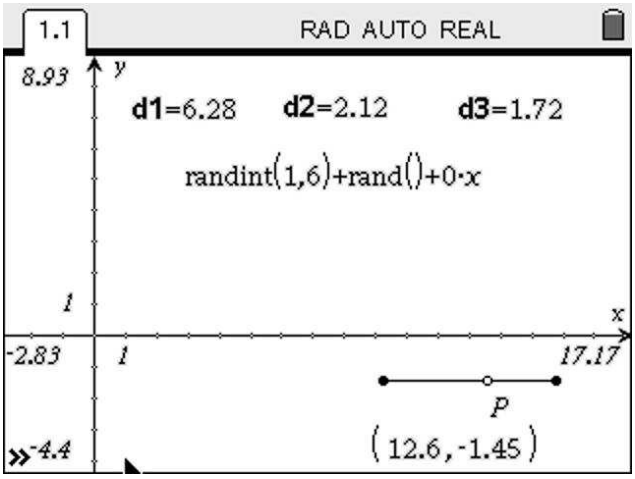
One thing one must always bear in mind is the final look of the simulator, so as to keep all objects as nice and tidy as possible, avoiding confusing screens.



Having built the control segment, it's time to generate the events. We write down the formula and calculate it three times (one for each die) using a coordinate of the control point as a silent parameter.

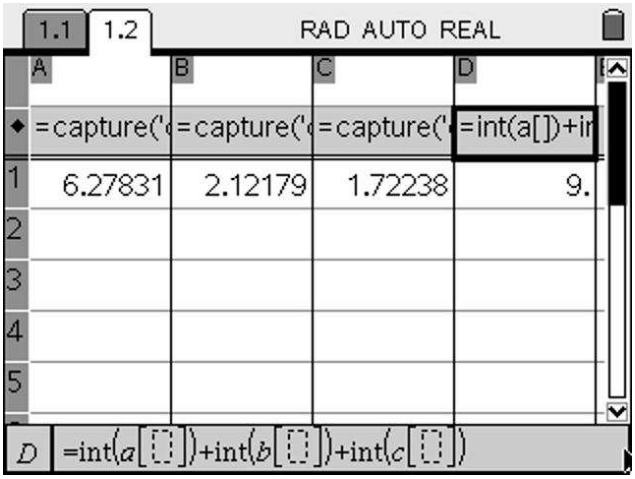
As we're simulating dice rolls we need the **randint** function to determine the score, plus the **rand** function to add noise to the results in order to capture them correctly.

Remember to store the results in variables.



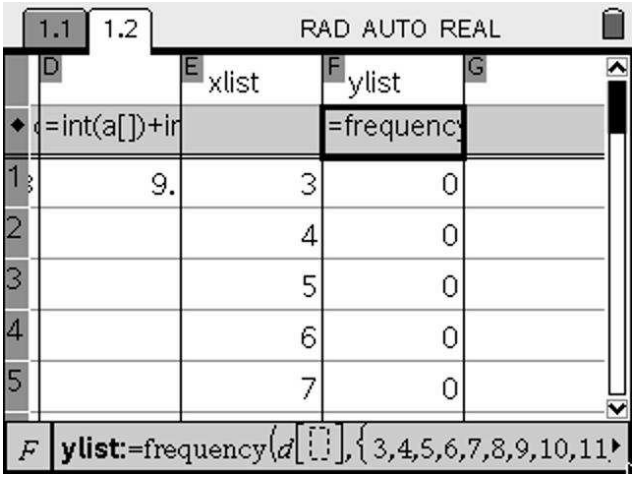
Now it's time to capture the results on a spreadsheet, filter out the noise (with the **int** function), and add them up.

Remark: as column D depends on the results of 3 different columns a "Dimension mismatch" error will occur when data is being reset. After reseting the data on all 3 column the error message disappears.



To get the final results, use the **frequency** function. Note that when determining the frequencies one must specify the bins and the resulting list will have +1 element, corresponding to the "None of the above" case. So, use a list going from 3 to 17 as the bins list.

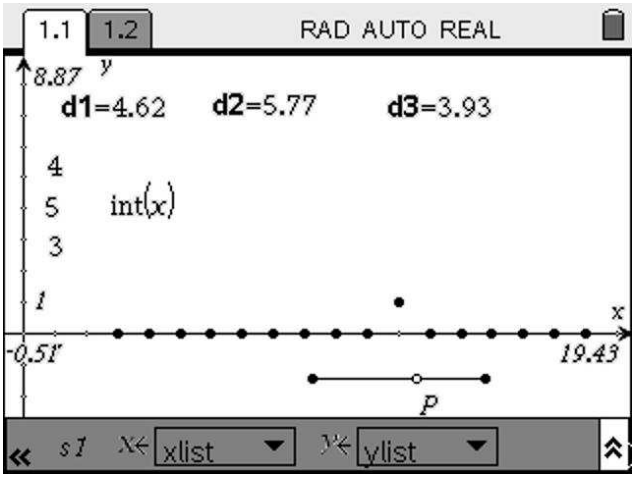
Name the columns to be able to display them as a scatter plot.



Back to the G&G page set up a scatter plot using xlist and ylist.

Round the results by calculating the formula **int(x)**, and adjust/hide everything.

Due to the window settings chosen this simulator won't be able to display large results; one option would be using relative frequencies, instead of number of occurrences.



Changing the animation attributes of the control point we now have a functional simulator that displays results in realtime. A lot of work still needs to be done, to make it look nice, fit the data to the window, etc., but the core of the simulator is ready.