Basketball Game | **Mini-Project 2: Draw the Net**

In this second mini-project, you'll draw the net using the **Line** command. You'll randomly generate a height for the next basket, so each time you play the net will be at a different location.

Objectives:

- import the code from the background file
- use **randInt** to generate a random height for the basketball hoop
- use Line to draw your net

Basketball Game Project Overview:

After completing a series of 8 mini-projects, you will have a basketball game similar to the one on the right. The code for projects 1 -4 will be imported into project 5. Projects 6-8 will build upon project 5.

After Project 6

Mini-Project Order:

1. Draw the Background
2. **Draw the Net**
3. Power Gauge
4. Angle Gauge
5. Merge the Projects and Code the Arrow Keys
6. Toss the Ball
7. The Game
8. Win the Trophy (and fireworks!)

The Net

1. Create a new program named NET

   You will import your BACKGRND code from the basketball project 1. We import this code to help troubleshoot bugs. If at any point your program becomes too hard to understand, you can start over and import the functioning BACKGRND code

   Use recall to import the background code

   Rcl (2nd sto→)
   prgm
   EXEC
   Select BACKGRND
   Press the enter key

2. You may either delete the last three lines. Then, retype them at the very end of the program. Or, you can insert several lines above the pause using the insert lines above command found in [alpha] [graph].

   :Pause
   :BackgroundOff
   :RecallGDB 1

3. To make the basketball hoop, you will need 6 line segments to make the rim. You will need to create multiple line segments for the net.

   Recall from project 1, the y values for the screen range from 1 to 165. The height of your hoop will be random. To start, you will let it be a random number from 30 to 100. You may adjust these numbers as you play with the game. To generate a random height, type

   :randInt(30,100) →Y

   You will need 3 more variables to store the various heights for your basketball hoop. Storing these values will make future coding in this and later basketball projects easier. The screenshot to the right demonstrates where these values will occur on your hoop. Write the three lines of code to create these variables.

```
NORMAL FLOAT AUTO REAL DEGREE MP
PROGRAM:NET
:165→Ymax
:BackgroundOn Image8
:DispGraph
:
:ClrDraw
:randInt(30,120)→Y
:Y-5→S
:Y-10→Z
:Y-25→N
```

4. To draw the yellow hexagonal rim takes 6 lines of code. Each line should be in the form

   :Line(X1, Y1, X2, Y2, Color)

   The top line in for the demo hoop is

   :Line(245, Y, 255, Y, YELLOW)

   The left side is:

   :Line(240, S, 245, Y, YELLOW)
   :Line(240, S, 245, Z, YELLOW)

   What are the three lines of code to create the base and the right side?

   :
   :
   :

5. The Net:

There isn't a required number of line segments for the net.

The demo picture has 8. Notice they aren't all slanted the same direction.

One sample line of code is:

```
:Line(240, Z, 250, N, DARKGRAY)
```

Code your line segments for your net. Execute your program often to check the location of your net pieces.

6. Execute your code several times.

Run the program several times to make sure that the position of the basket changes.Once you are satisfied your code functions properly, delete the ClrDraw from the top of the code. We won't need that line there for the rest of the projects.

If you deleted the three lines in step 3, add them to the end of the program.

```
:Pause
:BackgroundOff
:RecallGDB 1
```