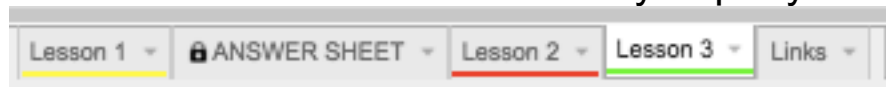


Monday 11/30/15 -- we continue our Scratch journey! In Lesson 3, we begin by creating a maze game and then we will learn about blocks. Blocks can be used for many things but they help us with fractals, one of the topics for this lesson. All homework is due by Sunday evening 12/13.

First Reminder: our Class Web page has TABS

Yes, the bottom of our spreadsheet at tinyurl.com/online4kids has TABS at the bottom. Each one is color coded. When you click on it, you will see the exercises for that lesson and that is where you put your comments!



Second Reminder: yes you should look at the work of other students!

Yes, I want you to check out how other students solve challenges and do various exercises. This is a wonderful part of Scratch!



Question #24: How do we change the ZERO POINT of a costume?
(checking that you read page 6 from Lesson 2)



Question #25: When was the first computer bug? Who found it?
(you will learn this later in Lesson 3 -- you don't know it now)

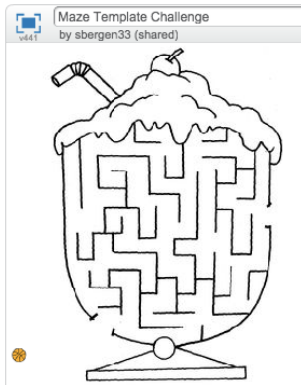
First Exercise Challenge: You need to make a Sprite travel in a maze game that you create!

I want you to get this MAZE TEMPLATE I have made for you that contains several maze choices. Then REMIX IT, SAVE A COPY and be creative. I am intentionally giving you freedom to be clever and create! You can do as many of my suggestions as you want or you can come up with your own creative ideas!

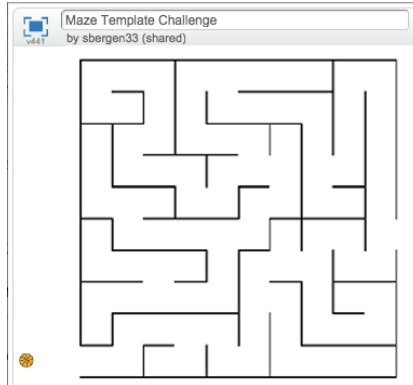
<https://scratch.mit.edu/projects/88875433>

(short name: tinyurl.com/mazetemplate33)

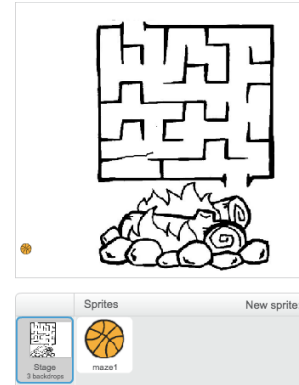
Exercise 3.1: Name your program "Exercise 3.1 My Maze Game" and make sure you share it. Send me a quick email when done so I can play it! There is no JV or Varsity for this 3.1 exercise.



or



or



Using the BROWN EVENT command that begins with WHEN lets you easily program the arrow keys to move in the 4 directions. Here are 2 of the 4 mini programs you will need. Of course you need an UP program and a DOWN program as well.



You can also CHANGE by a higher number if you want more speed since 1 goes pretty slowly!

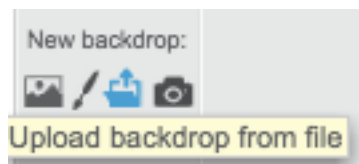


And yes, you can use the MOVE command instead, but this makes it harder when you want to go up and down so I don't like it!



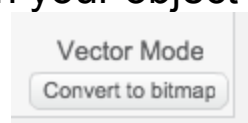
An Overview of Creative Ideas for this Exercise Challenge (you will have your own)

Choose any MAZE BACKDROP you want. I have given you 3. If you do a google IMAGE search for "mazes" you will find others. Once you find one you like, you "right mouse click on a PC" or "hold down CTRL and click on a Mac" to save the file to your computer, preferably the desktop. Then you UPLOAD it as your backdrop using the third icon at the bottom left of your screen.

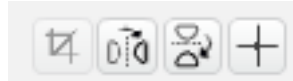


You can also choose any COSTUME you want. I have given you 4 but you can choose your own.

→ I put in costumes for a basketball, airplane, balloon and of course the Larry Bird Doll. You can choose any one that you want or add your own. However, if you add a COSTUME use VECTOR MODE in the graphics window (bottom right) to shrink the COSTUME to be small. VECTOR MODE means you can click on your object and resize by the corner.

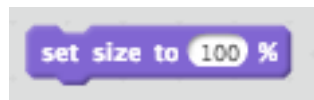


Also make sure the ZERO point is set at the spot you want using the GIANT PLUS at the top right of graphics window. I taught you about the ZERO POINT in Lesson 2.



I have set the ZERO POINT of the balloon, LB Doll, basketball and airplane to be in the exact middle.

You can also use the SET SIZE to 100% command to change the size of the costume.



Summary of Ideas (do as many as you want)

IDEA 1) program the arrow keys to go up, down, left and right -- perhaps program TURBO SPEED buttons that jump 10 up or 10 to the right to go faster

→ for example you can have a RIGHT ARROW that goes 1 and you can program the "R" BUTTON to go 5 or 10 to the right



IDEA 2) program a COUNTER that keeps track of the # of seconds it takes someone to complete the maze

→ for example, you can create two variables called COUNTER and SECONDS, setting them both to ZERO when the game begins. Then you can increase the COUNTER by 1 in a FOREVER LOOP, using the SET COMMAND and the ROUND COMMAND so that you can keep track of

the number of seconds as a whole number. I have estimated about 10000 counts in 1 second, but you might test this and be more accurate! The ROUND command takes the decimal of COUNTER divided by 10000 and rounds it to the nearest whole number.

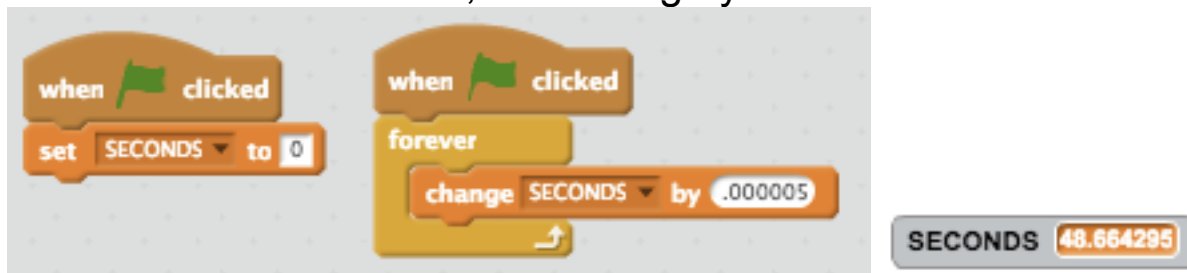


You don't have to display the COUNTER on your MAZE screen but you should display SECONDS using 1 of these 3 formats



You can change formats by holding down the SHIFT button and clicking on the icon.

Even more clever, you can create just one variable called SECONDS and have it in a FOREVER LOOP, increasing by .000005 each time.



You can make up a game where the user must complete the maze in under 100 seconds or the game is over. You can have a variable called SCORE which might be set to be 150 - SECONDS. You can then compete with others to have the highest score!

IDEA 3) use POINT IN DIRECTION command if you want your sprite to face up or face right when you change from up/down to left/right



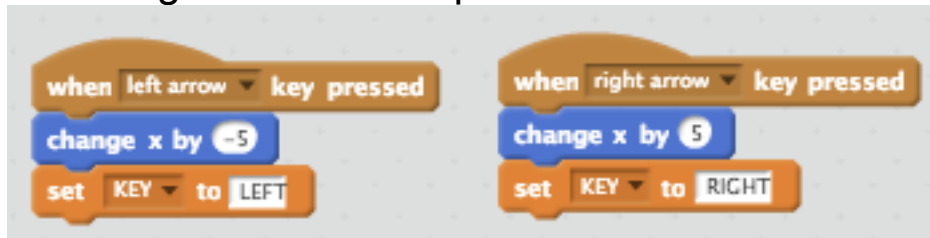
IDEA 4) program something to happen when the SPRITE hits a black wall
→ one choice would be to send the Sprite back to the starting spot. This might be cruel and unusual punishment but it is the easiest option to do!



→ another choice might be that the Sprite loses 3 seconds which affects the final score and you hear a sound



→ if you want it to "bounce off the black wall" when you hit then you need to have a variable e.g. KEY that keeps track of which KEY was pushed



Then when your sprite hits a black wall you can UNDO the move since you know the direction it was coming from. So if you pushed LEFT ARROW then when we hit the black wall, we compensate with right 5.



You need to implement it for each direction key you had in your code!

IDEA 5) program a SECOND SPRITE to randomly jump around the screen. I have put a WITCH into our TEMPLATE so you can use if you wish. Using the IF TOUCHING WITCH sensor command, you can make something cool happen when the WITCH randomly touches the ball. I have given you this code that makes the WITCH glide randomly:



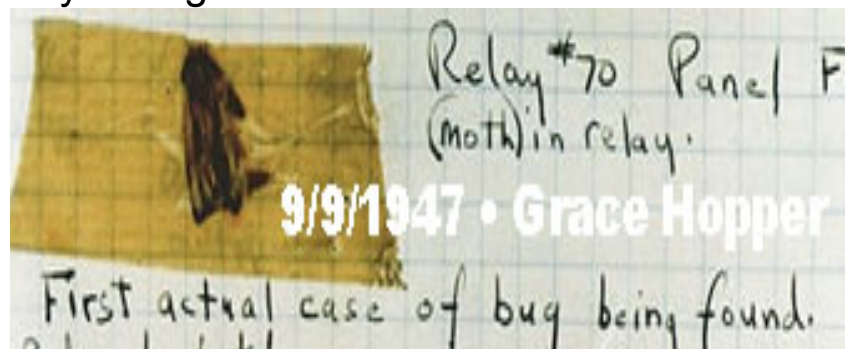
Please change this code as you like! I have also given you the code to decide what to do when the WITCH touches your main SPRITE.



What should you do when this happens? Perhaps BROADCAST a message to the WITCH sprite that makes it grow or shrink or make a sound? So you might have this code in your MAIN SPRITE:



and then in the WITCH SPRITE you might have



ENJOY • BE CREATIVE • HAVE FUN • WATCH OUT FOR BUGS!

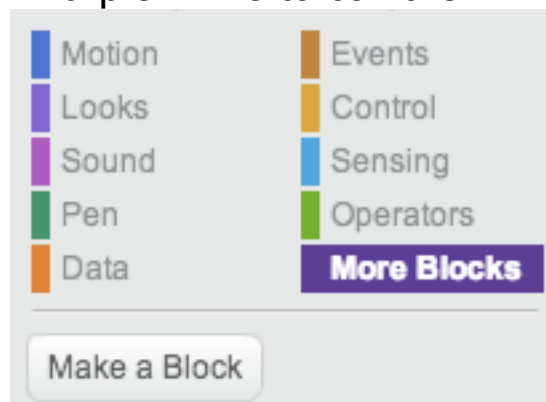
The first bug was in 1947 by U.S. Navy officer Grace Hopper who found a moth in a computer that was bigger than your living room! Serious! Feel free to google "the first bug" and read more about this story.

• above picture from: <http://thenextweb.com/shareables/2013/09/18/the-very-first-computer-bug/>

Next Topic: Let us Learn About Blocks (not bugs)

Yes Halloween is over but I tricked you when you created your time-consuming quiz in Lesson 2. So much repetition! I hope you duplicated the code for each of the 5 or 10 questions, but using MORE BLOCKS, there is indeed a better way!

If you look at the bottom right of the Scratch color choices, you will see it say MORE BLOCKS in Purple. I like to call them PURPLE BLOCKS.



When you make a new block, you are adding a new word to the Scratch language or defining a new procedure where some work will happen.

I can make a block to build a square of any specified size.

In this [first required 4 min video](http://youtu.be/P05EMU9sqSk) at <http://youtu.be/P05EMU9sqSk> or short name of tinyurl.com/spinning33 I am building a PURPLE BLOCK called SQUARE and using it to achieve Spinning Squares.

Or I can make a PURPLE BLOCK to create a new command. Or I can make a PURPLE BLOCK that handles the questions and answers for the 11 trick in lesson 2.



Question #25 again: When was the first computer bug? Who found it? (now you should know it)

Let's Make a Purple Block to Go Back

For example, we have the MOTION command



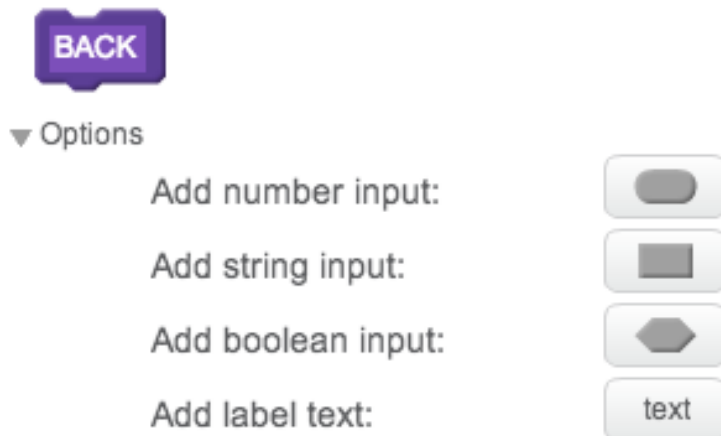
that we have used since our first lesson, but we do not have one that GOES BACKWARD 10 STEPS. Yes, we could do this



but pretend that this was impossible. So we use MORE BLOCKS to create one. When it says MAKE A BLOCK, we type BACK and then notice there are OPTIONS below.



When we click on OPTIONS, we get several choices, each with a purpose



The first 3 are called INPUTS and they allow us to "pass info" to the new block called BACK. For our purposes now, we want the first type (number input) so that we can issue commands like BACK 30 or BACK 140. Yes, I know we can MOVE -30 or -140 but I want to pretend we cannot and want to create this NEW command called BACK.

This **required 7 minute video** at <http://youtu.be/4pNpcn2R1Ck> or short name of tinyurl.com/blockback33 takes you through building the new COMMAND called BACK. It also has a section on the BACKPACK feature to reinforce that skill!

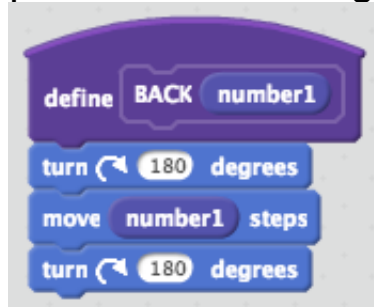
Boolean refers to True or False and that is yet another input, e.g. we could say RESPOND ($X > Y$) where $X > Y$ is either true or false.

And LABEL TEXT allows you as programmer to put some comments or notes about this block for your future reference or that of someone else if you are on a programming team. Remember that documenting your CODE is really important!

For our BACK block, we choose NUMBER TYPE and it will look like this:



We now put in the script for this command -- turning 180 degrees, moving the specified number of steps and then turning again.



and we now can go back to our main program and issue commands like



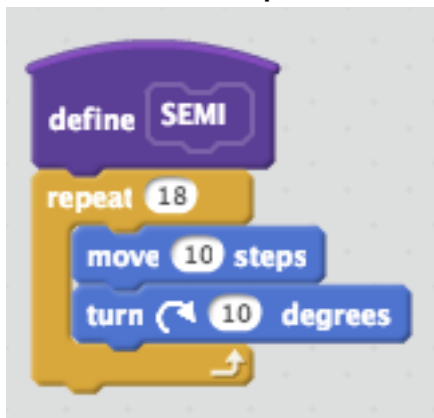
and everything acts as if Scratch has a new command!



Question #26: Purple blocks can be found from the main Scratch screen at what location? _____

Let's Make a Purple Block to Do A Semicircle

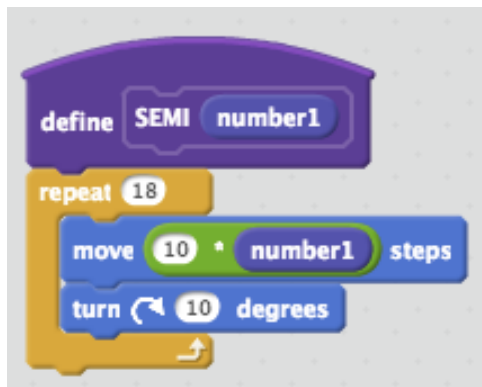
How about a block that draws a SEMICIRCLE which can be useful when drawing letters of the alphabet like "S" or "C" or "G"



which results in

Want to make the SEMI block fancier?

How about a number input that specifies whether the semicircle is bigger or smaller than one we just drew? We can EDIT the BLOCK called SEMI to have a number input and then use NUMBER1 as a multiplier for the number of steps.



so SEMI .5 is



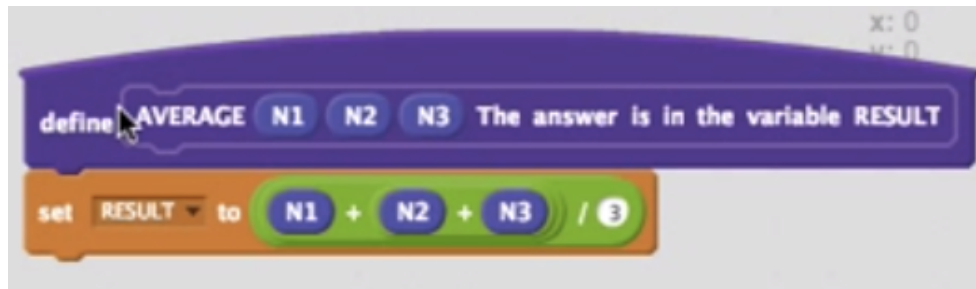
and SEMI 1.5 is



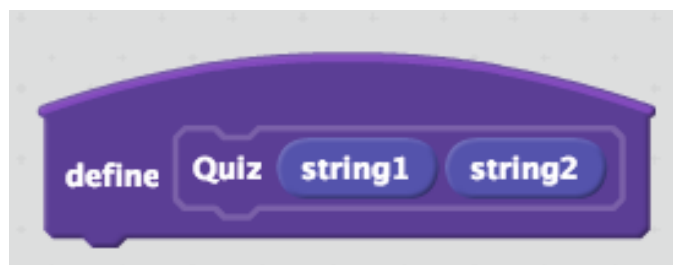
Let's Make a Purple Block to Find the Average

Reminder: to find the average of 3 numbers, you add and divide by 3.
So the average of 7, 13 and 40 is $7+13+40 = 60$ divided by $3 = 20$.

We can create an AVERAGE BLOCK that automatically finds the average of several numbers. Here is a [required 5.5. minute video](http://tinyurl.com/online4kids) on building the block called AVERAGE -- http://youtu.be/l4Qzw3j_TqE -- using three number inputs. Hopefully this video will solidify your awareness of BLOCKS.

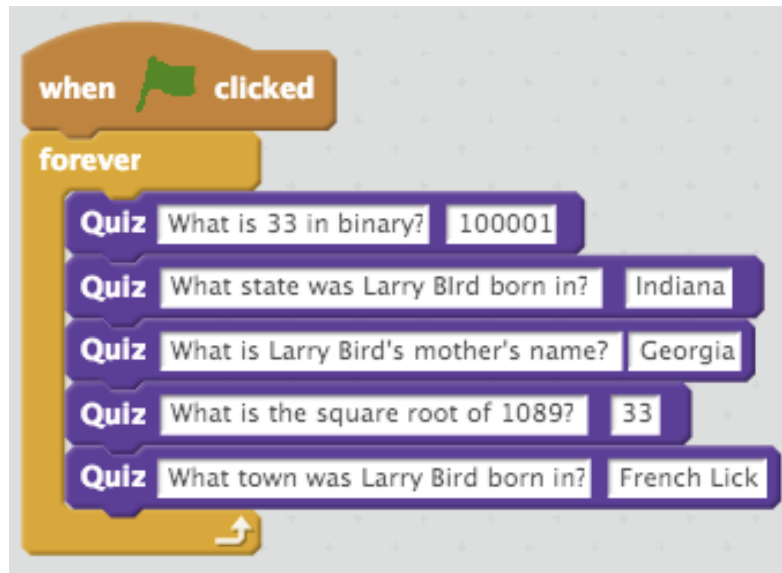


Finally Let's Make a Purple Block to Do a Quiz



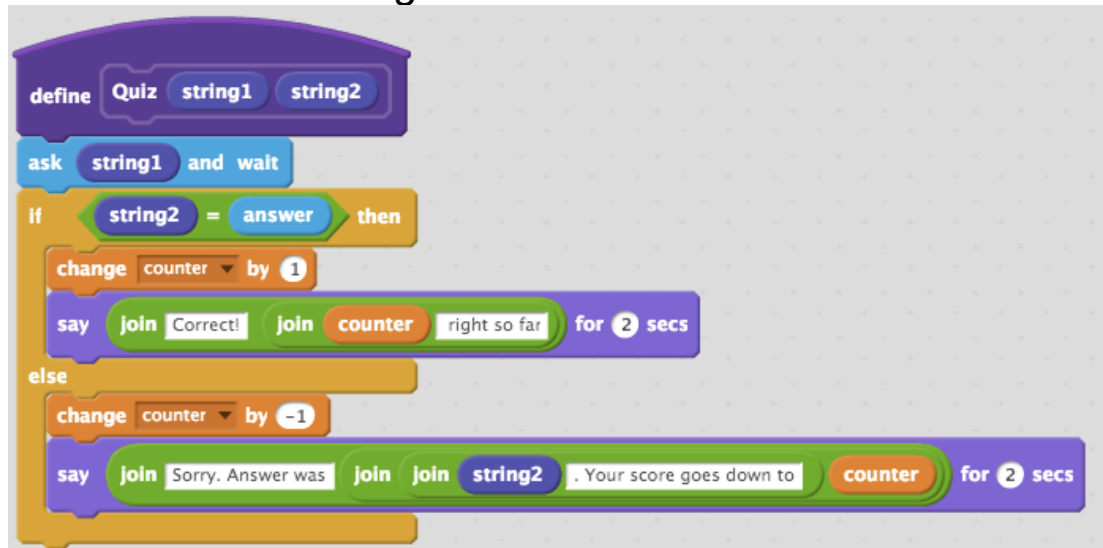
The two variables STRING1 and STRING2 are called PARAMETERS. We are passing these parameters from the main program to the BLOCK and saying "hey Block, please take charge of asking the question called STRING1 to the user and evaluate what he or she responds based on the answer I am giving you in STRING2."

One more point of advocacy regarding BLOCKS is that BLOCKS allow you as the programmer to keep your code shorter and more efficient. Efficiency in coding is a plus.



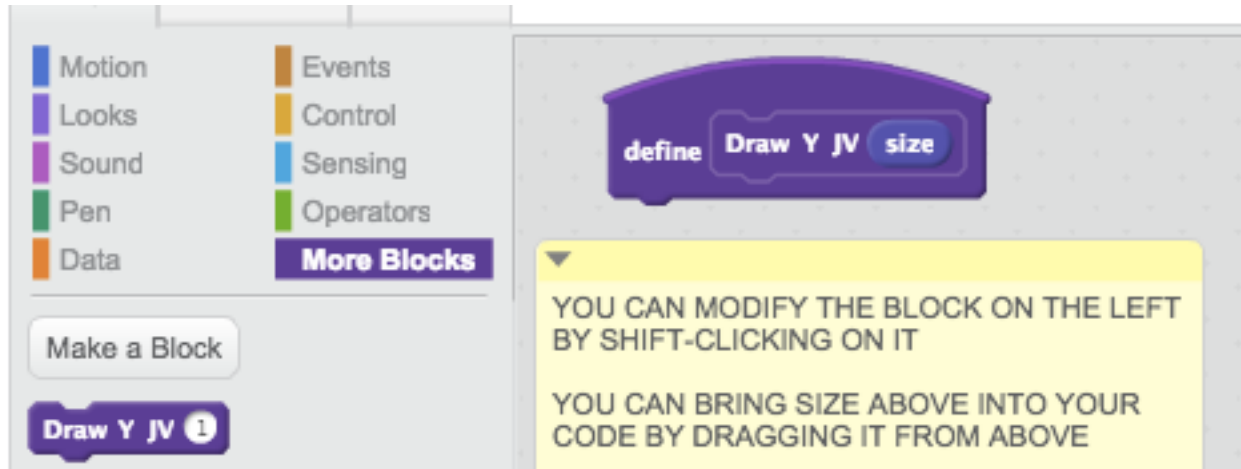
Here is a quiz program where I have my five questions. The block called QUIZ takes care of all the work. In other computer languages, what we call blocks in Scratch might be called SUBROUTINES or PROCEDURES.

Here are the details of what goes on within the block called QUIZ.



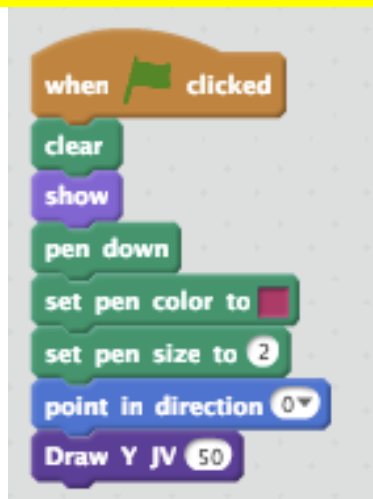
Notice how the computer asks the question called STRING1 of the user and then evaluates whether STRING2 matches ANSWER, either adding or subtracting a point from the user's score via the variable COUNTER.

In Lesson 4, I will ask you to make ANOTHER QUIZ program -- this one with purple blocks.



Second Homework Challenge: Make a "Y" Block

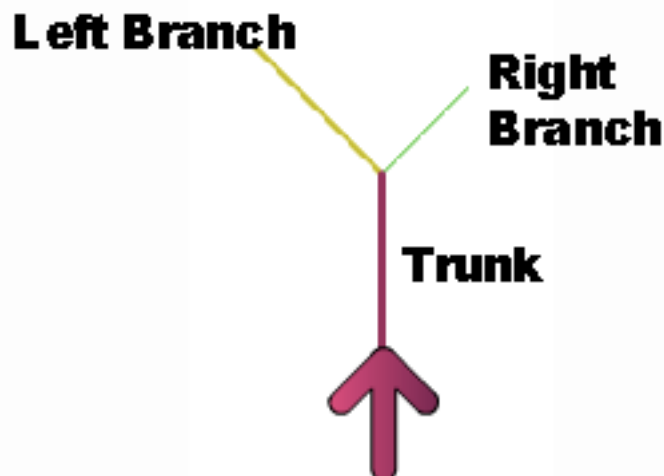
JV Exercise 3.2J: Create a PURPLE BLOCK called DRAW Y JV that draws an UPPERCASE Y of specified size. Name your program "Exercise 3.2J Draw a Y Block" and make sure you share it. Use code on left below as your GREEN FLAG code that calls the PURPLE BLOCK. Change the 50 to other numbers and test, test, test. Your sprite must come back to where it started! Do not use GO TO (,) in your code for this Purple Block!



JV solution if you need help is at <https://scratch.mit.edu/projects/88984345>
or short name of tinyurl.com/branchtree33

Varsity Exercise 3.2V: Same as JV but we want the LEFT BRANCH of the "Y" to be a different color and 3/4 the length of the TREE TRUNK. Name your program "Exercise 3.2V Draw a Y Block" We want the RIGHT BRANCH of the "Y" to be a different color and 1/2 the length of the TREE TRUNK. We want the RIGHT BRANCH to be PEN SIZE 1, the TRUNK to be PEN SIZE 3 and the LEFT BRANCH to be PEN SIZE 2. Your sprite must come back to where it started! Do not use GO TO (,) in your code for this Purple Block!

Varsity solution if you need help is at <https://scratch.mit.edu/projects/88989565> or short name of tinyurl.com/branchtree34



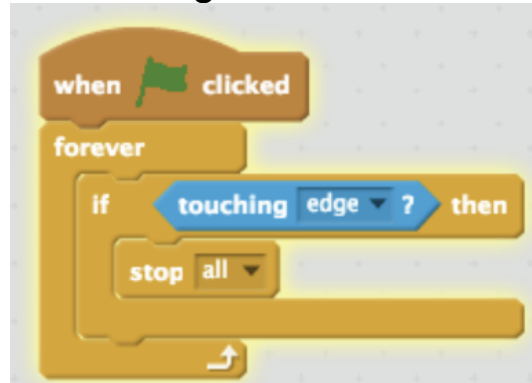
Question #27: True or False: You are allowed to use GO TO in doing JV Exercise 3.1J or 3.1V? _____

Question #28: What is 11 x Larry Bird? _____
Hint: Larry's uniform number was 33

Question #29: True or False:
Efficiency when coding is a good thing _____

Next Topic: Tessellation with Borders

In Lesson 2, you all did either 2.2J (square tessellation) or 2.2V (hexagon tessellation). When you get to the border of the stage you compensated with this code from lesson 2 using the SENSING command:



Now I want to show you the clever trick for making a tessellation finish!

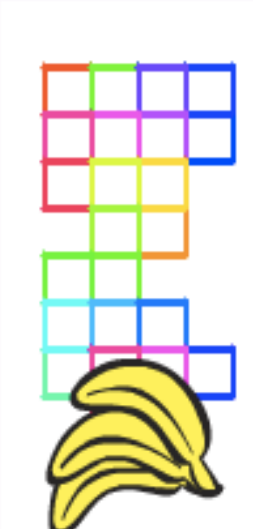
Create a second program that checks IF TOUCHING EDGE
But instead of STOPPING, the way we did in Lesson 2, we use this strategy.

- PEN UP
- MOVE BACK to ORIGIN (0,0)
- CHANGE ORIENTATION TO SAME AS ORIGINAL
- PEN DOWN AND TESSELLATE AGAIN

By the way, another word for STRATEGY is PROCEDURE and a fancier word is ALGORITHM.

The word ALGORITHM means a strategy or procedure to solve a problem.

On the next few pages, I am going to show the steps -- the algorithm -- using the colorful tessellation of squares by LRL from WI. The existing program follows the logic of lesson 2 and stops when it hits the edge.



```

when green flag clicked
  pen down
  set pen color to 199
  set pen size to 3
  go to x: 0 y: 0
  point in direction 0
  clear
  show
  say BANANA for 2 secs
  forever
    repeat (pick random 4 to 8)
      move 20 steps
      turn 90 degrees
    turn 90 degrees
    change pen color by 10
  
```

```

when green flag clicked
  forever
    if touching edge? then
      stop all
  
```

done by LRL from WI
 strategy IF TOUCHING
 EDGE



Question #30: True or False: The variable in a PURPLE BLOCK can be dragged into multiple locations in the code below. _____



Question #31: True or False: The name of the brown MESSAGE you use with the brown BROADCAST command can be your choice. _____

The first thing we need to do is create a message and separate the code that draws the tessellation. This is easy. Check this out please! I used the BROWN EVENT command so I could chop the code into two pieces. The top half ends with BROADCAST and the bottom half begins with WHEN I RECEIVE.



```
when green flag clicked
  pen down
  set pen color to 199
  set pen size to 3
  go to x: 0 y: 0
  point in direction 0
  clear
  show
  say BANANA for 2 secs
  broadcast draw square

when I receive draw square
  forever loop
    repeat pick random 4 to 8
      move 20 steps
      turn 90 degrees
    turn 90 degrees
    change pen color by 10
```

done by LRL from WI

strategy IF TOUCHING EDGE

separate the SQUARE and use a MESSAGE to connect

Test it!

Now we change the algorithm for what happens when it hits the edge.
Look at the code below in the upper right.



done by LRL from WI

strategy IF TOUCHING EDGE

separate the SQUARE and use a MESSAGE to connect

Test it!

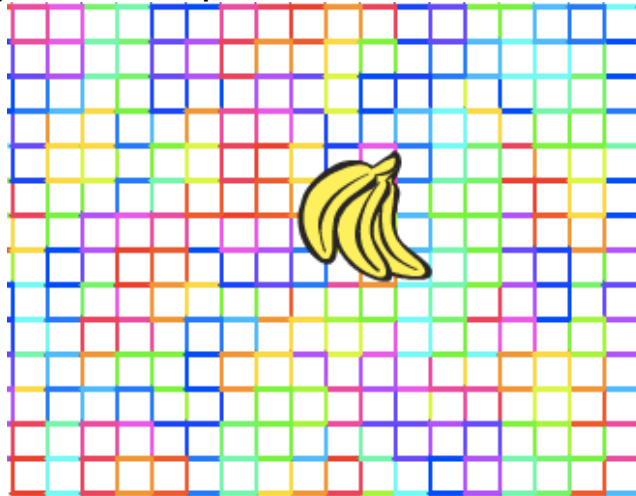
Now change the IF TOUCHING so that the sprite lifts the PEN then moves to the CENTER ORIGIN then POINTS IN DIRECTION 0 and then resumes drawing the TESSELLATION



Question #32: True or False: The SHIFT button lets you click and change the name of an orange VARIABLE or a brown MESSAGE name or a purple BLOCK. _____

Here is the result filling up the entire screen with the square tessellation.

You can see the code at <https://scratch.mit.edu/projects/88915427>
or short name of tinyurl.com/squaretess35



JV Exercise 3.3J: Watch this required 7 minute video at <https://youtu.be/TI3I128avok> or short name of tinyurl.com/squaretess34 -- and then modify your Square Tessellation from Lesson 2 using SAVE A COPY so that it becomes 3.3J. Name your program "Exercise 3.3J Square Tessellation Improved" and make sure you share it.

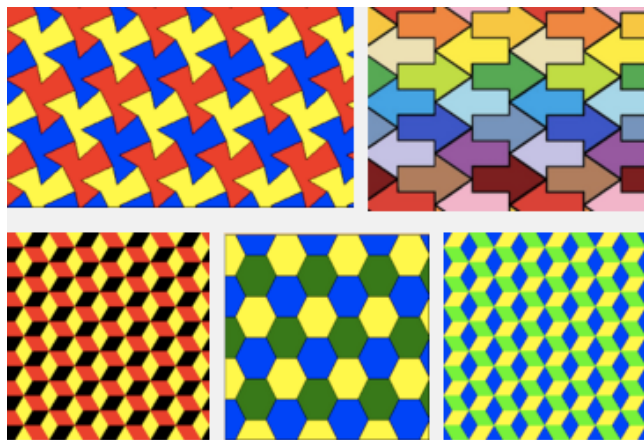
Varsity Exercise 3.3V: Watch this required 11 minute video at <https://youtu.be/QjvyVHlwj7E> or short name of tinyurl.com/hextess34 -- and then modify your Hexagon Tessellation from Lesson 2 using SAVE A COPY so that it becomes 3.3V. Name your program "Exercise 3.3V Hexagon Tessellation Improved" and make sure you share it.

Super Varsity Challenge 3.3S You must first do 3.3J or 3.3V. Then watch this video at https://youtu.be/E_NpPaU8dJQ or short name of tinyurl.com/polygonfill33 -- now make your tessellation fill in the shapes using the algorithm I show you on the video. Here is the algorithm -- *INSTEAD OF GOING FD X STEPS, use a REPEAT X LOOP and DRAW LINES ONE AT A TIME TO THE RIGHT*

Next Topic: Making Your Own Fractal

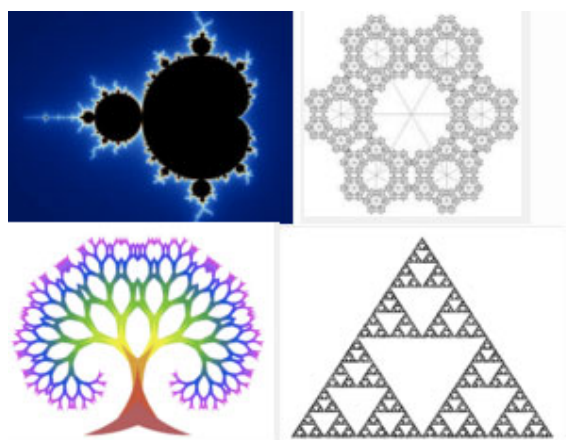
First let me explain what a fractal is in terms of comparing to a tessellation. In a tessellation, all the repeating shapes in the pattern are the **same** size. In a fractal they get smaller and smaller, proportional to the original -- usually with a nice pattern and semblance of order.

TESSELLATIONS



Do you see each one is made of 1 single shape or perhaps 2 shapes but they don't change their size?

FRACTALS



Do you see there is a big shape and then a smaller version of that big shape and then a baby version of that same shape.

So my challenge is to use the Lesson 2 program of variable sized letter 2.1J or 2.1V to make our first fractal in this course. It won't be as beautiful as the fractals above but it will be a fractal.

Exercise 3.4J: Name your program "Exercise 3.4J My First Fractal" following instructions in the video and PDF. Make sure you share it.

Exercise 3.4V: Create it using a PURPLE BLOCK. Study my sample at bottom of page 27. Name your program "Exercise 3.4J My First Fractal" following instructions in the video and PDF. Make sure you share it.

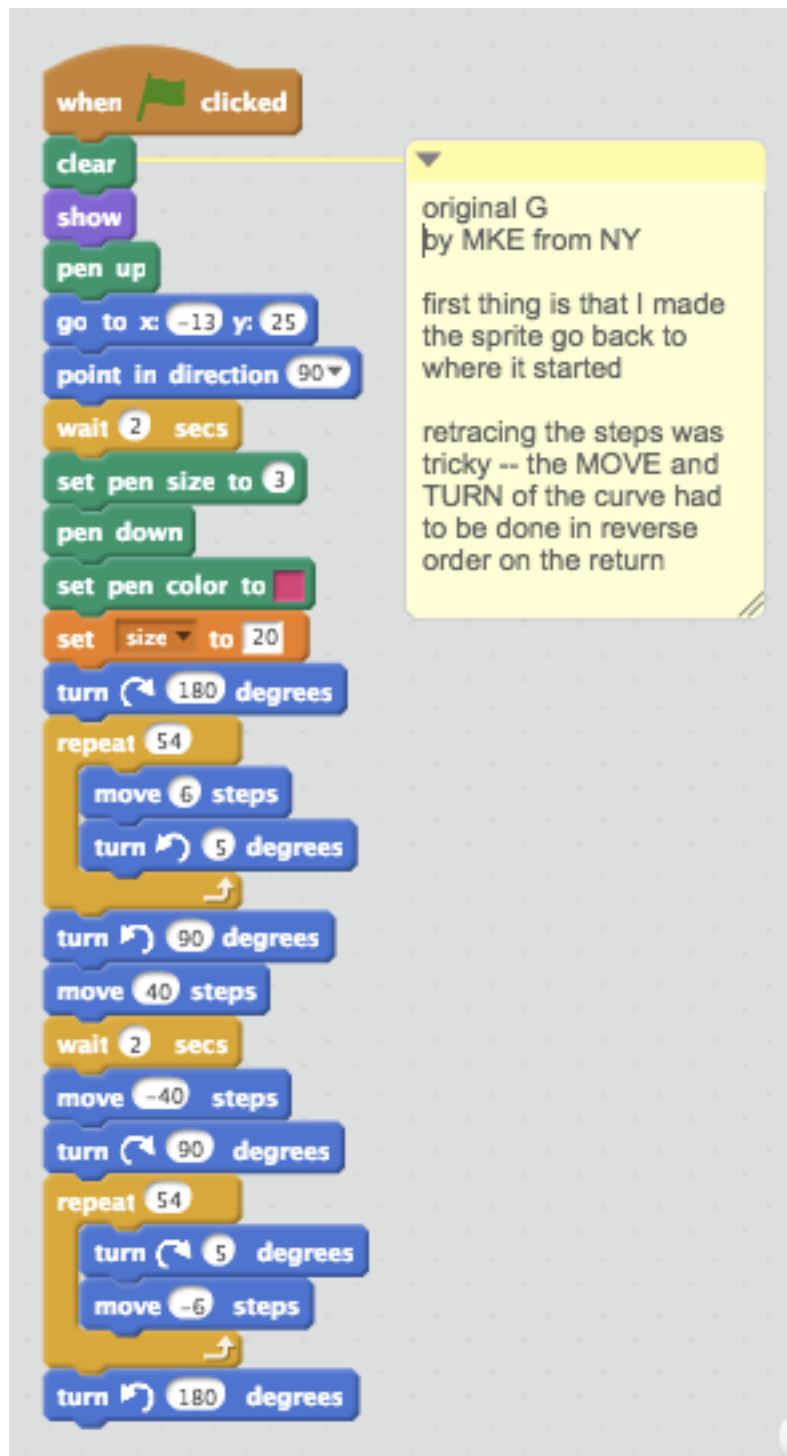
Both JV and Varsity -- please start by watching this video where I make a FRACTAL using the work of WJC from TX. <https://youtu.be/0vRtmdlKd7E> or short name of tinyurl.com/fractalT33

On the next few pages, I show you a sample program -- step by step that draws a G. I chose to use the work of MKE from NY to get started. Notice the clever way of using a partial circle with the REPEAT 54.

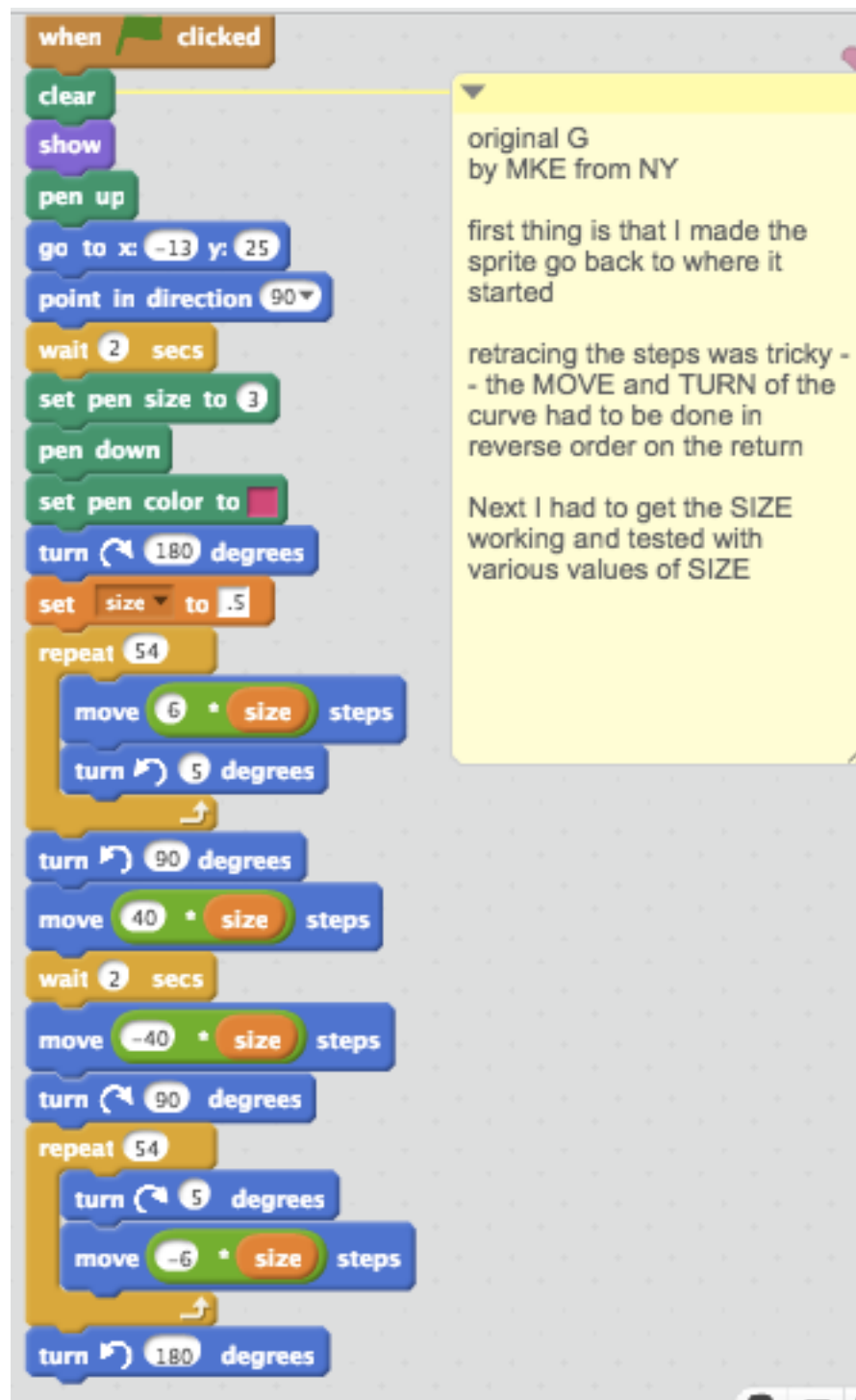


So there were a number of things I had to do to get this G ready to be a fractal. These may well be the things you need to do.

First thing I did but it is not essential-- I made the sprite return back to the original spot and original angle. I do that out of habit since it is a useful technique when stringing a bunch of letters. In the code below you will see me "backtrack" on the G so it gets back to where it started.

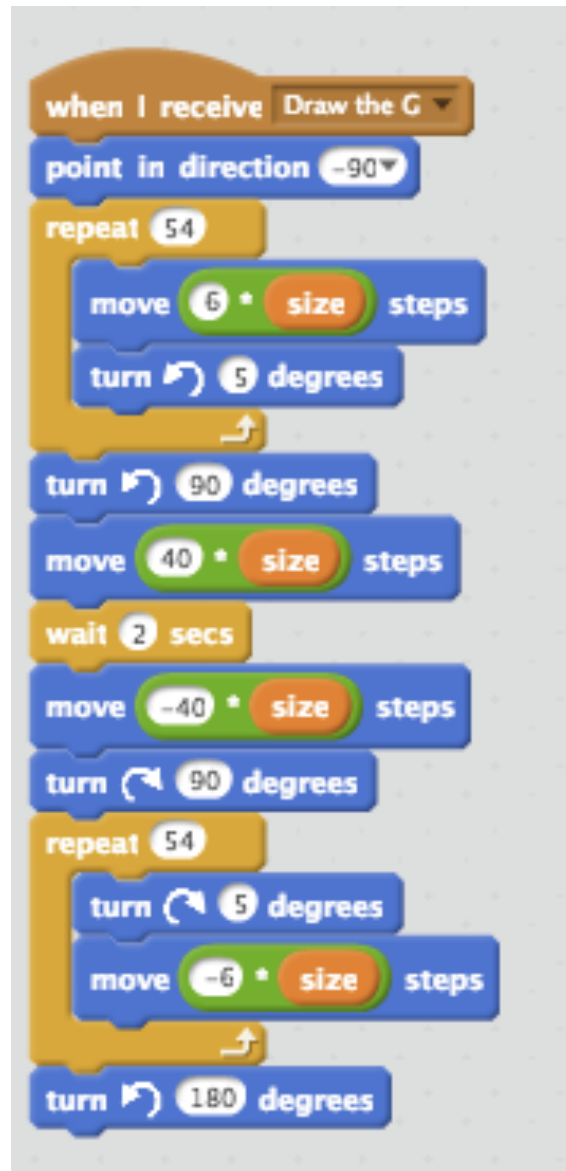


Next I have to make the variable called SIZE work by applying it to each of the MOVE ____STEPS command. I like to do this one step at a time and test, test, test. If you change 5 or 10 commands and then test, it is very frustrating and hard to find your bug.



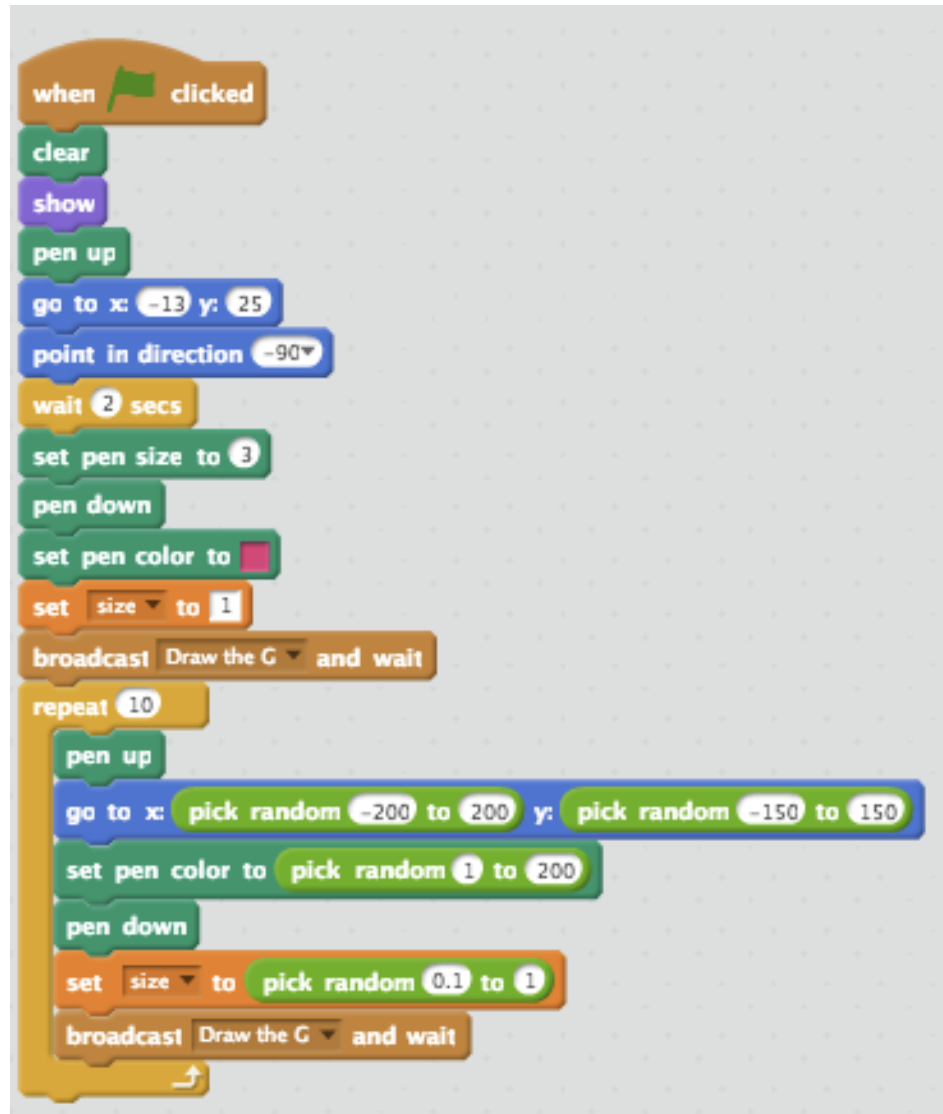
Moving the sprite back to where it came is not important for the JV challenge. If you do the Varsity with BLOCKS then it is important!

Now that I have tested the code and it works, I created a brown EVENT MESSAGE called DRAW THE G. The reason for this is so I can do lots of G's over and over again in different sizes -- my fractal idea!



Notice how the SIZE variable is there for each and every MOVE as a multiplier.

My main program now clears the screen and draws 1 large G followed by 10 smaller G's of random size from .1 which is 10% to 1 which is 100%



Below are a few different outcomes of my G fractal.



You can see my fractal G at <https://scratch.mit.edu/projects/88912203>
or tinyurl.com/fractalg

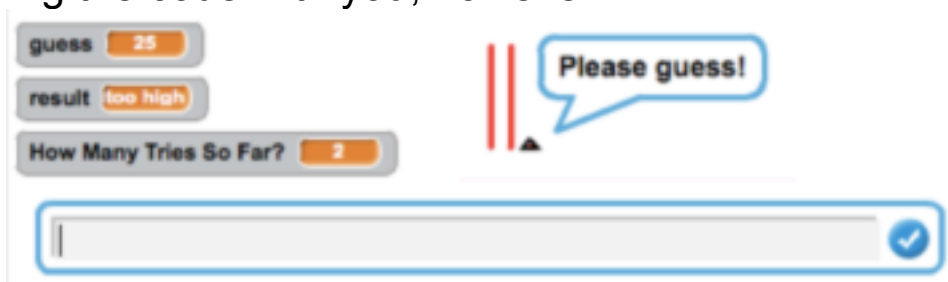
VARSlTY Exercise 3.4V I also modified the code using PURPLE BLOCKS (on right above). You can see this code at

<https://scratch.mit.edu/projects/89429114> or tinyurl.com/33purple

SUPERVARSITY 3.5 Challenge: COMPLETELY OPTIONAL

Write a program that has the computer pick a number from 1 to 100 but not tell you. The computer then lets you guess and tells you each time "too high" or "too low" until you guess the number. Name your program "Exercise 3.5SV Too High Too Low Game" and make sure you share it.

If this interests you and you choose to work on this one, please watch the optional 2 minute video at <http://youtu.be/NCUcEcZxxeM> or short name of tinyurl.com/guess34 that explains it all. Sample screen snapshot below. I am not sharing the code with you, however!



SUPERSUPERVARSITY 3.6 Challenge: OPTIONAL of course!

Write a program to find the ONE NUMBER under 2000 that is equal to the SUM OF TWO SQUARES in 2 different ways.

$$A^3 + B^3 \text{ or } C^3 + D^3$$

where A,B,C and D are 4 different whole numbers above 0. This is very hard! Name your program "Exercise 3.6SSV Special Number" and make sure you share it. Email me if you try it and I will help!

Hint: This is a famous math problem and I will give you the name if you try it. All 4 numbers are below 15, for example $2^3 + 13^3 = 7^3 + 8^3$

END OF LESSON 3 -- ALL HW DUE by SUNDAY NIGHT 12/13