Project Ideas

Below is a list of project ideas. You should feel free to do something else that is not on this list. Anything is good as long as it is interesting to you and you get practice using variables, conditionals, and for loops (and maybe even arrays)! If you finish your project, you can always try adding more and more features to it! You can also discuss them with me to get ideas on what to do next.

Scratch

Hangman Implement Hangman, where a random word is selected and the user has to guess what letters are in the word. When the player guesses a correct letter, fill in that letter in the mystery word. If they guess a letter that is not in the word, then they lose a life. If the player loses a certain number of lives before guessing the word, they lose! Start by having a Sprite say the word, replacing unknown letters with '?' and having a sprite slowly disappear as you get wrong guesses. Once you have this, try using Letter Sprites and Clones to display the game.

Bouncing Ball Start with a ball sprite moving horizontally. Have it bounce back off the edge of the screen. Add a vertical component and have the ball reflect off walls (e.g. if it hits the right wall at 30 degrees, it bounces off the wall at -30 degrees). On the bottom of the screen, add a paddle the user can move left and right to hit the ball. If the ball hits the bottom of the screen, end the game. Count how many times the player hits the ball! Make the game more complex: add a second vertical paddle on the right side of the screen. Add multiple balls and end the game when any (or all!) balls hit the edge of the screen. Make the game speed up

Typing Training Design a program that helps you practice your typing skills! Start with a letter sprite that is gray. When the user presses that key, have the color return to the letter. Add several letters, and only return color if it is the next letter in the word. Add a timer and display the user's Words Per Minute at the end of the game! To extend this project, you will want to use the `clone` feature to make this more advanced: start with gray A - Z off screen, and clone the letters of a randomly chosen word and have the clone appear on screen. Use a "For This Sprite Only" `cloneID` variable to treat multiple clones of the same letter differently (i.e. so only the right clone re-colors when it should)! Use a list to pick different words/sentences to type.

Simple Game Make a little game! For example, you could have your character Sprite navigate a 2D town and talk to other sprites. Your character could have a goal (e.g. get a certain item), and have to trade items and do simple tasks for NPCs.

MakeCode

Tamagotchi Design a virtual pet for the micro:bit! Feed it, play with it, and take it for walks.

Cheat Codes One of the most famous video game cheat codes is the **Konami Code**:

$$\uparrow\uparrow\downarrow\downarrow\leftarrow\rightarrow\leftarrow\rightarrow \texttt{BA}$$

Implement the Konami Code on the micro:bit. If the user inputs the code, print a secret message! Once you've implemented one cheat code, try implementing several!

Building on Tutorials There are many tutorials on the micro:bit MakeCode website. Follow one or two lessons and extend their functionality!

Java

Drawing

This section has a few projects that use StdDraw.java. Most of these are mathematics based. If any of these sound interesting to you but you're not quite sure about the math, I am more than happy to explain them in more detail!

Chaos Game Draw an equilaterial triangle and choose a random point within the triangle, marking it with a small dot. Pick a random vertex of the triangle and place a dot half way between your current point and that vertex. Repeat this 10000 times, using the newly plotted dots as your new starting place. Done correctly, you should generate a Sierpiński Triangle. For an extra challenge, expand your program to start with an n-gon and to place the dot $0 \leq r \leq 1$ of the way to the chosen vertex. Wikipedia has some example values for n and r that produce interesting fractals.

Fractal Snowflake Start with an equilaterial triangle. At each iteration we will do the following: on each line segment, draw an equilaterial triangle whose base is the middle third of that line segment. Remove the base of the newly drawn triangles (i.e. the middle third of the line segment we started with). The resulting figure is known as the **Koch snowflake** and is an example of a fractal. Create a `Snowflake` program that draws the nth iteration of the Koch snowflake. Allow the program to take a command line argument to decide which iteration to do. Implement other variants listed on Wikipedia.

Mandelbrot Set Start with a point (x_0, y_0). We define the point (x_1, y_1) so that $x_1 = x_0^2 - y_0^2 + x_0$ and $y_1 = 2x_0y_0 + y_0$. We define $(x_2, y_2) = x_2 = x_1^2 - y_1^2 + x_0$ and $y_2 = 2x_1y_1 + y_0$. In general, define $(x_n, y_n) = x_{n-1}^2 - y_{n-1}^2 + x_0$ and $y_n = 2x_{n-1}y_{n-1} + y_0$. If the distance from (x_n, y_n) to the origin, draw a blue dot at (x_0, y_0), otherwise draw a black dot. Do this for several thousand random starting points in the circle of radius 2 centered around the origin. You will generate the Mandelbrot Set, and as you set n bigger and bigger, you will get a better and better picture.

Collatz On your HW, we saw the Collatz sequence, where each odd number is subsequently multiplied by $3n + 1$ and each even number is divided by 2 until the sequence reaches 1. While Mathematicians have checked that this sequence reaches 1 for starting at every number up to at least 2.36×10^{21}, it still hasn't been proven to always reach 1. Create a Collatz program that checks the numbers from 1 to n and prints out how many steps each starting number takes to reach 1. Print these to a file. Use StdDraw.java to create a graph showing how many steps each starting number takes to reach n.

APIs

During our Networking lesson, we learned how to call an API in Java. Find another free API and design your own console app like we did for the weather. Some possible APIs include:

1. MBTA
2. NASA
3. Star Wars

A list of free APIs can be found here. You can also improve the app from the activity (e.g. provide different data, provide a forecast for tomorrow, make a program that tells me if I need an umbrella, etc.)

Miscellanious

Doomsday Want a neat party trick? There's an algorithm for "easily" determining what day of the week a date falls upon. You can learn about it here. Create a game where you are given a date and need to reply with which day of the week it fell on. If you get it wrong, have the program print the Doomsday calculation!