Programming
SNAKE!

Summary

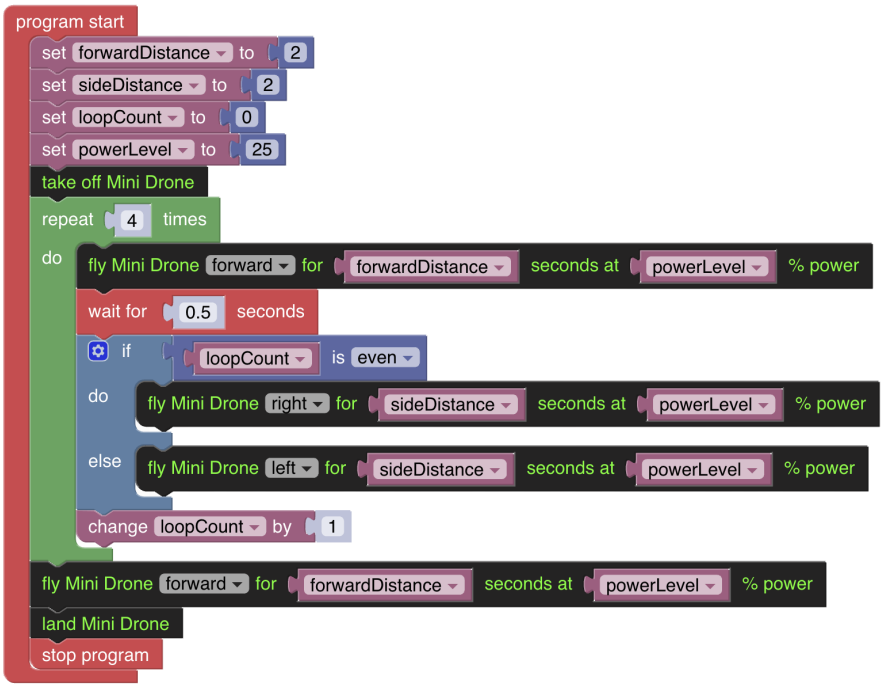
1. Subject(s): Coding (Block Code)
2. Objective: To design a autonomous program using loops, variables, and functions
3. Materials Required: Parrot Mambo drone, Bluetooth enabled device
4. Time Allotment: 1hr

Implementation

Learning Context

Snake is a classic video game from the late 70s. The basic goal is to navigate a snake and eat as many apples as possible without touching the walls or the snake's body. Similarly, In this lesson we will program the Parrot Mambo drone to follow a snake-like path that incorporates loops, variables and functions to achieve this.

Procedure



Finished Code

Javascript:

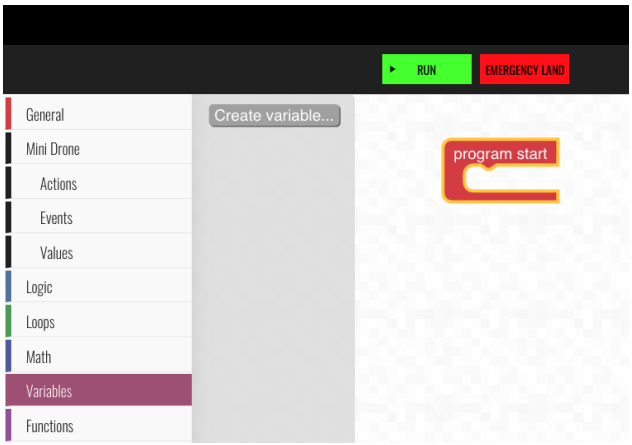
```
var forwardDistance, sideDistance, loopCount, powerLevel;


startProgram = async () => {
  forwardDistance = 2;
  sideDistance = 2;
  loopCount = 0;
  powerLevel = 25;
  await drone.takeOff();
  for (var count = 0; count < 4; count++) {
    if(--window.LoopTrap == 0) throw "Infinite loop.";
    await drone.fly("forward", forwardDistance, powerLevel);
    await drone.wait(0.5);
    if (loopCount % 2 == 0) {
      await drone.fly("right", sideDistance, powerLevel);
    } else {
      await drone.fly("left", sideDistance, powerLevel);
    }
    loopCount = (typeof loopCount == 'number' ? loopCount : 0) + 1;
  }
  await drone.fly("forward", forwardDistance, powerLevel);
  await drone.land();
  stopProgram();
};
```

In this lesson, we will be using a few variables to determine our drone's distance, speed and the number of loops that will determine our snake pattern. Variables are a great way to have consistency throughout programming and also makes for efficient ways to change variable data all at once.

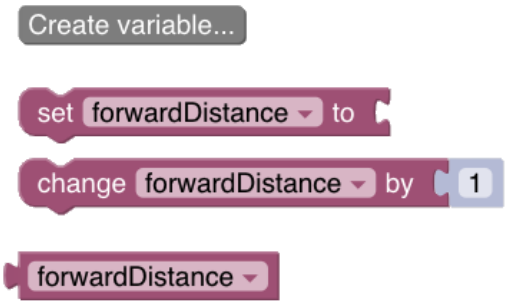
The first step in our snake program is to create a *variable* for our forward distance. In FTWCODE, our forward distance is determined by **direction**(forward, backward, right, and left), **time**(seconds), and **power**(percentage).

CREATING VARIABLES

To create a forward distance variable, we will begin by placing a **Program Start** block and then heading to the **Variables** blocks and selecting **Create variable**.



Name this variable **forwardDistance**. This variable will be used to determine the amount of time (seconds) the Mambo will fly forward.



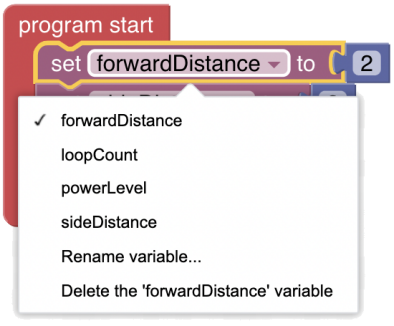
(Variable naming is an important aspect in making your code readable. Naming variables follow a simple idea: Create variables that describe their function and follow a consistent theme throughout your code. In this lesson, we will be following the Camelcase convention. In Camelcase, words are delimited by capital letters, except the initial word, e.g., flyingDrones.)

Create another variable called **sideDistance**. Similar to **forwardDistance**, this variable will be used to determine the amount of time the Mambo will fly left or right.

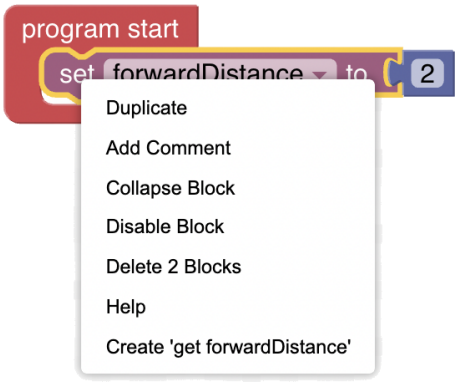
Next, we will be creating a **powerLevel** variable to determine the amount of power applied to the Mambo's motors.

Lastly, create a variable called **loopCount**. The **loopCount** variable will track the amount of times that the Mambo moves left or right within a loop. This variable is how we create the snake-like side movement.

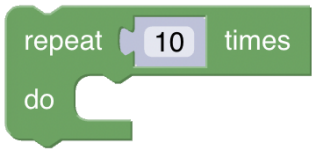
With the variables created, we can now begin to construct our code. In order to initialize our variables, we must first set them in our beginning code. This is done by grabbing the **set blocks** in the **Variables** block set. We then set our variable amounts using the **Integer** block in the **Math** block set. Using the drop-down menu, select the appropriate variable for each block.



In order to make the code writing process process more efficient, the duplicate block feature can be used. By right clicking on the block and selecting Duplicate. Blocks can also be deleted using the drop-down menu.

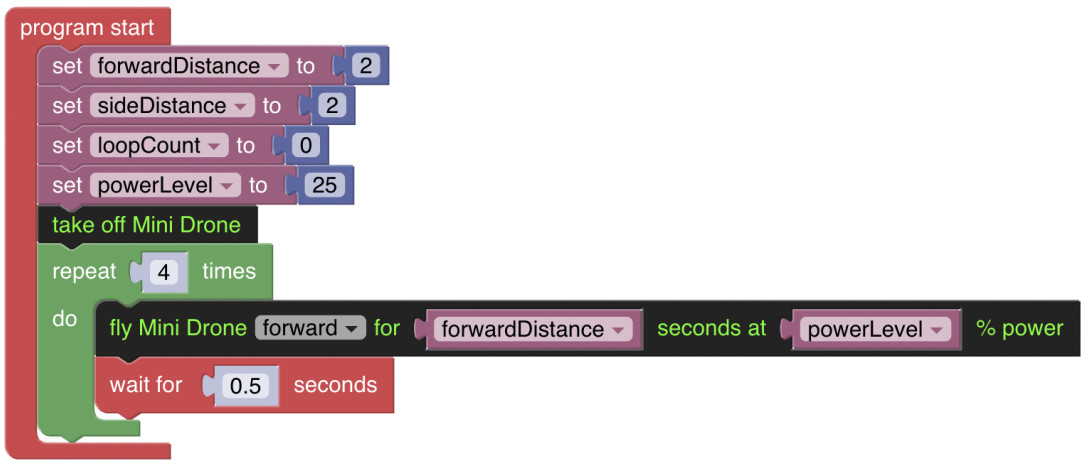


LOOPS



Once the variables are initialized, a **take off Mini Drone** command can be added. This command will launch the Mambo vertically and into a hover position. Following the take off command, a **repeat (loop)** block will be added. **Repeat** blocks are located in the **Loops** block set. Since the SNAKE! Program is relying on counting the number of times commands are run within the **repeat** block, we must use the **repeat times** block. Change the value from **10** times to **4** times for this exercise. (This value can also be adjusted depending on the amount of space available.)

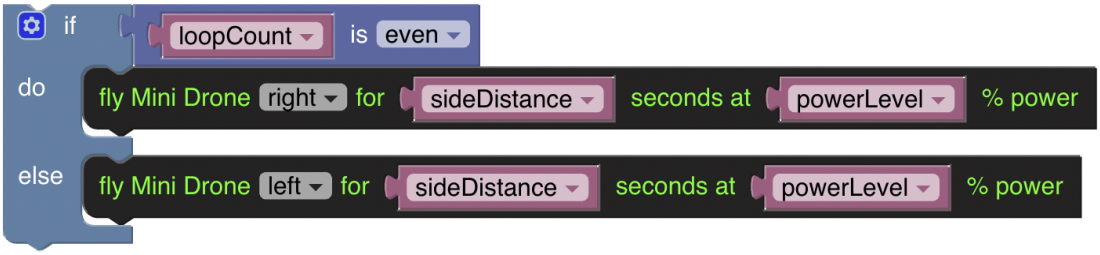
Add a **fly forward** command and a **wait** block inside of the loop. Change the default **wait** value from **1** second to **0.5** seconds. The **wait** block allows the Mambo to hover for a defined amount of time. We will also be replacing the default values within the **fly forward** command to utilize the variables (**forwardDistance** and **powerLevel**) created earlier. Grab and drag the variables from the **Variables** blocks into the corresponding seconds and power.



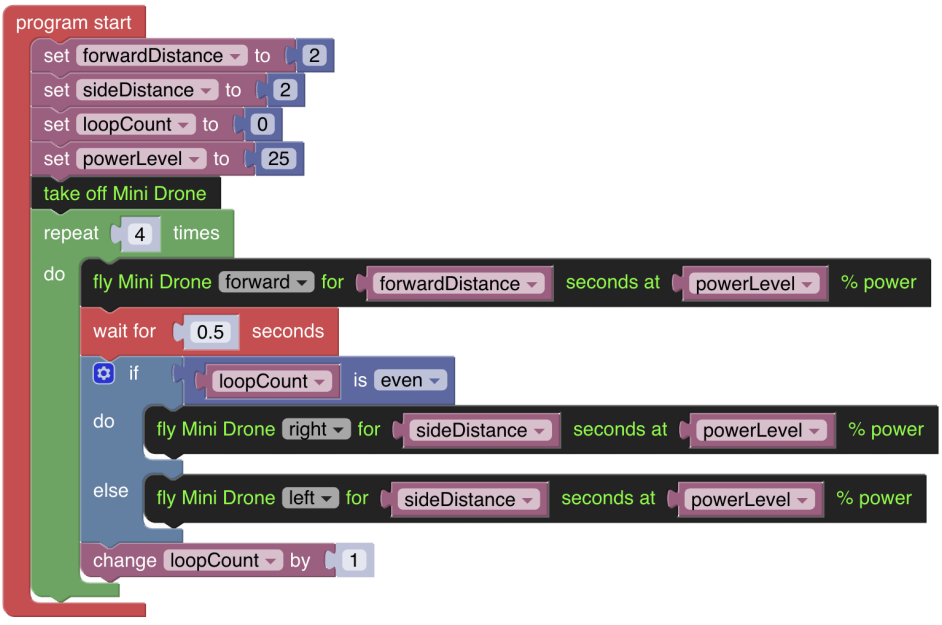
CONDITIONALS

After the **wait** block, we will use a conditional statement to determine the amount of times the Mambo will snake (turn). In this case, an **if then else** statement (located in the **Logic** blocks) will be used and utilize our **loopCount** variable, which was originally set to **0**. In the **if then else** statement below, our statement is determining flight pattern(**left** and **right** movement) through the validation of the condition set through the boolean value, **even**. Booleans are value types that have two possible values: true or false. Most often, boolean values are located in the **Math** block set.

If the **loopCount** value is **even**, the Mambo will fly **right** for the **sideDistance** amount (**2** secs) set and at a **powerLevel** amount (**25%** power). If the **loopCount** value were to be **odd**, the Mambo will fly **left** for the **sideDistance** amount (**2** secs) and at a **powerLevel** amount (**25%** power).

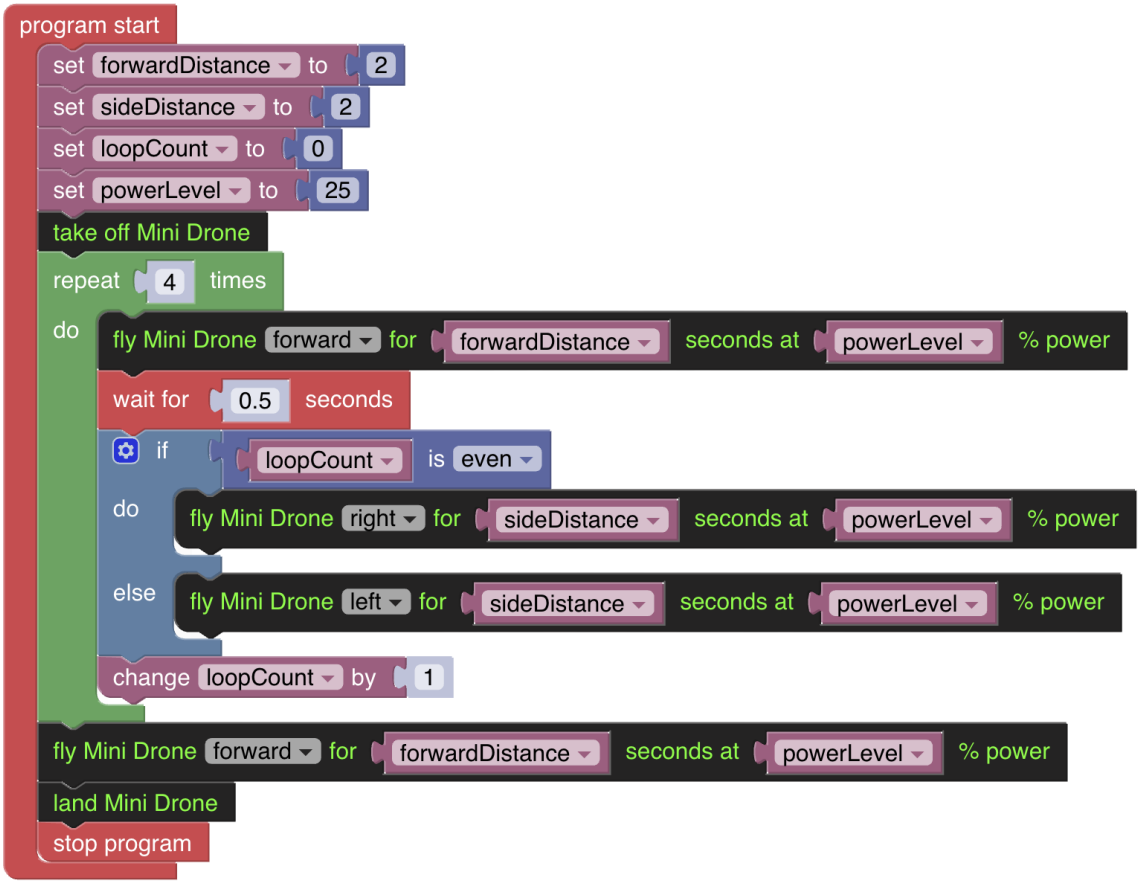


Knowing this, our current code would keep the Mambo flying right since there has been no condition set to change the value of **loopCount**. In order for the Mambo to fly in both directions, the **loopCount** value must change between **even** and **odd** values. To accomplish this, insert a **change variable** block and change the default value to **loopCount** and set the integer to **1** (default value).



Since **change loopCount** is within the **repeat** block, **loopCount** will increase by **1** until the repeat block has reached its specified value. In this instance, the code within the **repeat** block would loop **4** times. The **loopCount** value would also be 4 at the end of the program.

Finally, to end the SNAKE! program, a **fly forward** command and a **land Mini Drone** block should be added outside of the loop. A **stop program** command can also be added to guarantee that the code and Mambo are completely stopped at the end of our code.



The final SNAKE! program should look like the code above. As with all coding, modifications can be made and we encourage you to modify and play with the values within. The above code can also be found here and is ready to be imported into FTWCODE.

Materials & Resources

a. Instructional Materials: Mambo Drone, Bluetooth enabled device
b. Resources:
    a. What is a Variable? (CS Discoveries: Variables Part 1)
    b. Programming Basics (Statements & Functions: Crash Course Computer Science #12)
    c. Intro to Programming: Loops (Codeacademy)
    d. Conditionals: If and If/Else Statements (Computer Science Fundamentals, CODE.org)