

Micro:Bits

Micro:Bits Bop-It!



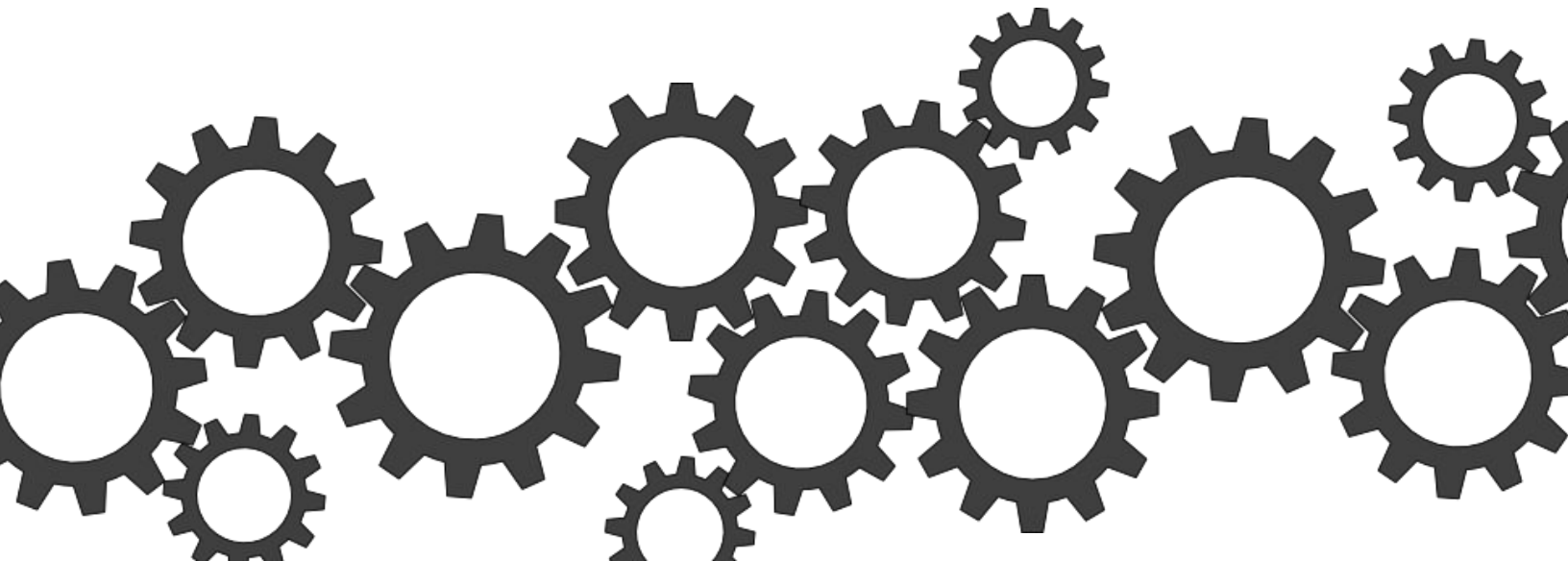
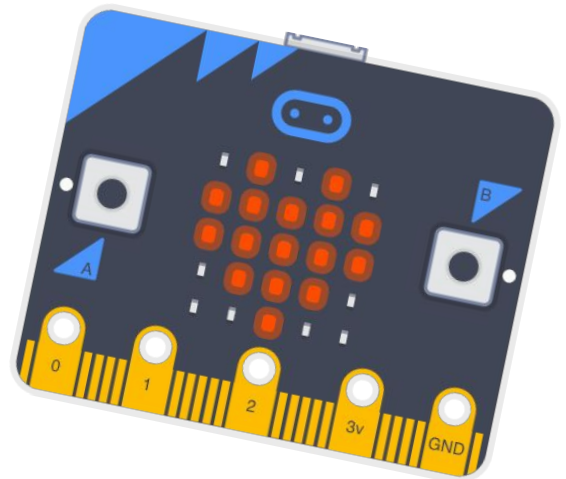
Turn your Micro:Bit into a Bop-It style game! Remember what action to take for each picture that the Micro:Bit shows you.

What is a Micro:Bit?

A Micro:bit is a miniature computer with a 5x5 LED display grid. We can use the MakeCode editor to code a program to execute on our Micro:bit. A program is simply a set of instructions for our Micro:bit to complete.

Materials


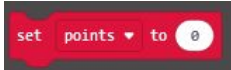
- Micro:bit kit
 - 1 Micro:bit
 - 1 Battery pack
 - 1 Micro USB cable
- Optional: 1 Dongle adapter if using Mac



Micro:Bits Bop-It!



Procedure

- 1) First, let's make sure we understand the intended functionality of the game. We want to recreate the original Bop-it game, but instead of calling out the actions, we will display the action through the LED screen. Let's start by getting the game working with two different actions and reactions, then we can add more. When the LED displays a ghost, shake the Micro:bit, and when it displays a butterfly, press the B button. The game will run for a set number of rounds (let's start with ten), displaying a random image each round, and you score 1 point for each correct action. So, the maximum score will be ten if you complete all of the actions correctly. Once we are happy with how the game runs, we can add rounds with increasing speed.
- 2) Let's initialize our variable for keeping track of the score first. Under the Variables tab, click  and name it "points". Then, pull out the  block to put in the "on start" frame. Great! Now each time the program starts, you start out with zero points.
- 3) Next, let's set up the main loop for the program. Go to the Loops tab, and pull out the loop frame, seen below. Set the number to 10 to start. The inside of this loop will make up one round, so we will have ten rounds total.



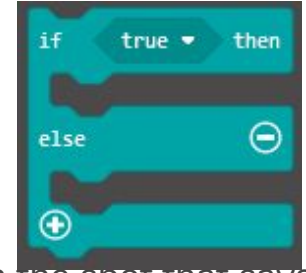
- 4) Now make another variable called "action". We will use this to keep track of which action needs to be done that round. Pull out the "Set action to 0" block and put it inside the loop. If we had to do the same action each round that would be too easy! So, go to the Math tab, find the "pick random" block, and put it in the set action block. Change the numbers to go from 1 to 2, since that gives us two options for an action. So far:




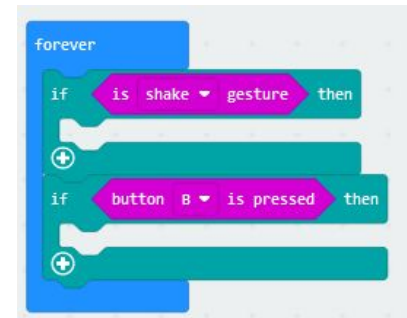
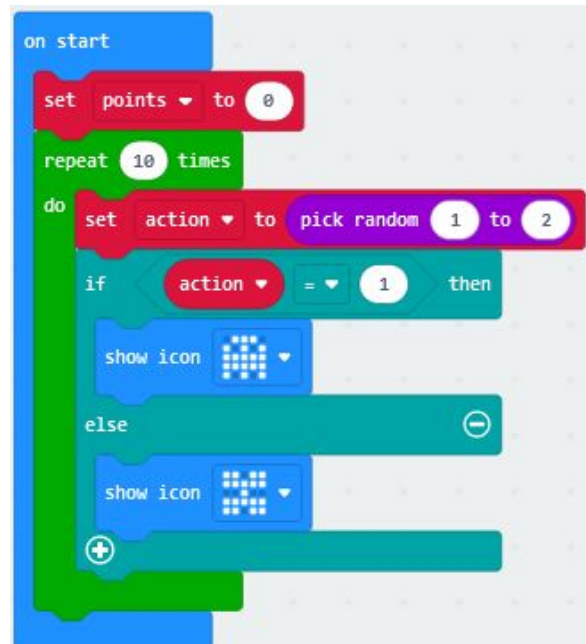
Micro:Bits Bop-It!



- 5) Now, what happens next depends on whether 1 or 2 was chosen. So, we need to bring out an if-else block (also known as a conditional). Go to the Logic tab, and pull out:



Also from the Logic tab, bring out  and put it in the spot that says "true". Find the rounded action block in the variables tab to put in one side, and replace the other side with a 1. Then, pull out the icon you want for the first action (I'm using the ghost), and put the second icon you want in the else loop.



- 6) Notice we don't need an "action = 2" condition, since if the action was not 1, it had to be 2.
- 7) Now, we need to find a way for the program to check if you are doing the correct action, and give you a point if you do. One way to do this is to have a forever loop running which stores the most recent action completed by the player. First, go to Basic and pull out a Forever frame. Each time through the forever loop we want to check if the Micro:bit is being shook, or if the B button is being pressed. So, go to the



Logic tab and bring out two  frames. The conditions we will put in

these blocks are from the sensors, so go to the Input tab & bring out



and  for the two frames. So far:

Micro:Bits Bop-It!



8) We will have to keep track of the player's action, so we can pass that information on to the "player action" variable. In most cases, we would use a "set player action" block, but this works fine for our purposes. We will set player action to 1, and if

we will use a "set player action" block, so that we can make a variable called "player action". We cannot have spaces in the variable name. In the "is shake gesture" block, set player action to 1. In the "button B is pressed" block, set player action to 2.

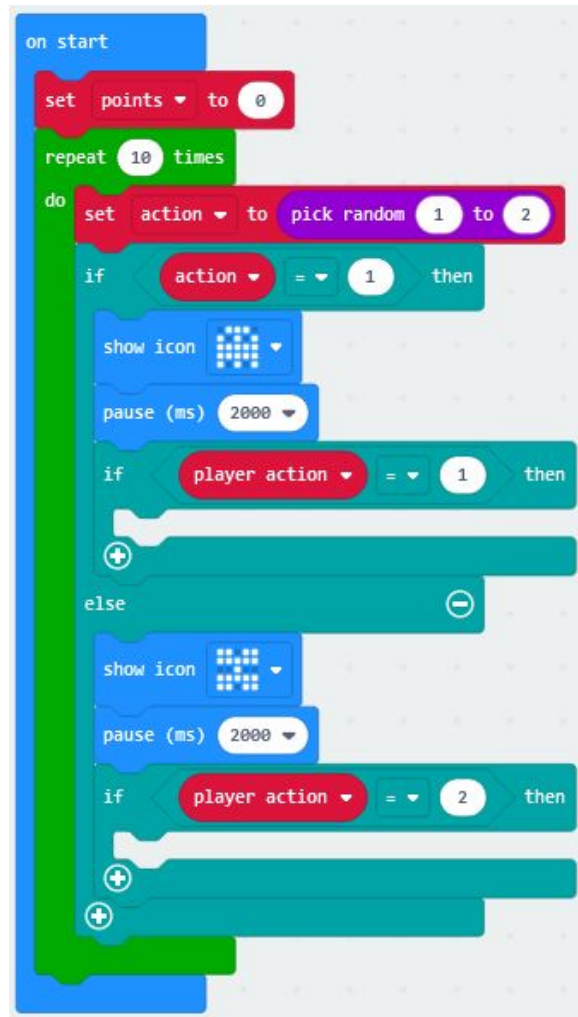
9) This block keeps running in the loop under "on start" and checks for the current action. In our "on start" loop, let's check

been made by the player, and give them a point if that is true. First, bring out a pause block to put under each of our action icons (so that the player has time to respond!). Let's start it off as 2 seconds (2000 milliseconds) for testing purposes.

Micro:Bits Bop-It!



- 10) Drag out another “if” block from the Logic tab and put it under the first pause block. Bring out the “ 0 = 0 “ condition from the Logic tab as well. From the variables tab, bring out the rounded “player action” block for one side of the condition, and enter 1 into the other side of the condition. Duplicate this frame to put under the second pause, but enter “player action = 2” instead.



- 11) These conditions are only satisfied if you did the correct action! So, inside of these blocks we will increase the score and show a check mark, so we know we did the right response. From the variables tab, pull out the “change ____ by 1” block, then change the variable to be “points”. Then, bring out “show icon □”.
- 12) After the green repeat loop, add in a “show string” and a “show number” block. Write “score” or something similar in the string block, then bring “points” from the Variables tab for the show number block. Let’s stop and test our games now!

Micro:Bits Bop-It!



```
on start
  set points to 0
  repeat 10 times
    do
      set action to pick random 1 to 2
      if action = 1 then
        show icon [grid icon]
        pause (ms) 2000
        if player action = 1 then
          change points by 1
          show icon [grid icon]
        else
          show icon [grid icon]
          pause (ms) 2000
          if player action = 2 then
            change points by 1
            show icon [grid icon]
      else
        show icon [grid icon]
        pause (ms) 2000
        if player action = 2 then
          change points by 1
          show icon [grid icon]
  show string "score"
  show number points

forever
  if is shake gesture then
    set player action to 1
  if button B is pressed then
    set player action to 2
```

Micro:Bits Bop-It!



- 13) The game doesn't work! What gives? The issue is that the "Forever" frame does not start to run until the "on start" frame has finished. Luckily, this is an easy thing to fix, and it also allows us to add a couple nice features to the game. Go to the Basic tab and bring out another Forever frame. Then, click on the "set points to 0" block, and drag it into the forever frame (this will drag all of the code blocks below it with it). We can leave the "on start" frame empty for now. Your new program will look like this:

```
on start
  set points to 0
  repeat 10 times
    do
      set action to pick random 1 to 2
      if action = 1 then
        show icon [3x3 grid]
        pause (ms) 2000
        if player action = 1 then
          change points by 1
          show icon [4x4 grid]
        else
          show icon [2x2 grid]
          pause (ms) 2000
          if player action = 2 then
            change points by 1
            show icon [3x3 grid]
      end if
    end do
  end repeat
  show string "score"
  show number points
```


Micro:Bits Bop-It!



- 14) This introduces a new problem: each time you complete a game, it returns to the top of the loop, sets your score back to zero, and starts the game all over again. Let's add a "start" button, so that you can control when you start the game. Drag out an "if" frame from the Logic tab, and put it at the top of the forever loop. Then, drag the "set points to 0" block into the if statement, which should bring the rest of the code with it. From the Input tab, bring out the "button A is pressed" condition, to put into the conditional we just placed. This is a good time to stop and test the game.

```
forever
  if button A is pressed then
    set points to 0
    repeat 10 times
      do
        set action to pick random 1 to 2
        if action = 1 then
          show icon [grid]
          pause (ms) 2000
          if player action = 1 then
            change points by 1
            show icon [grid]
          else
            show icon [grid]
            pause (ms) 2000
            if player action = 2 then
              change points by 1
              show icon [grid]
        show string "score"
        show number points
```

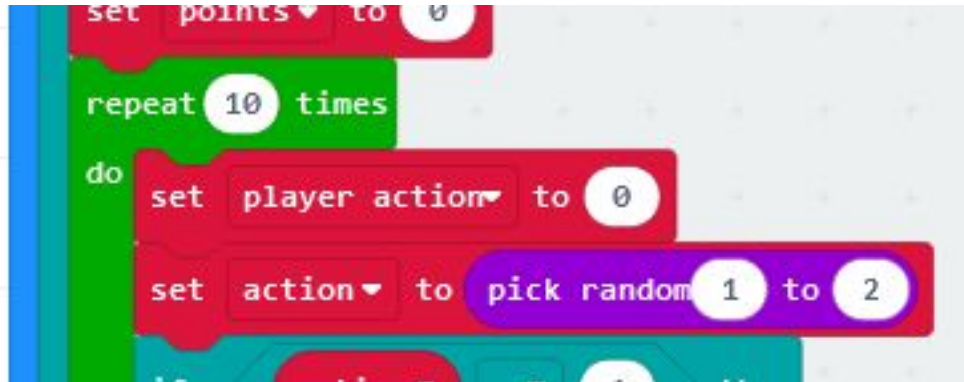
```
on start
  key
```

```
forever
  if is shake gesture then
    set player action to 1
  if button B is pressed then
    set player action to 2
```

Micro:Bits Bop-It!



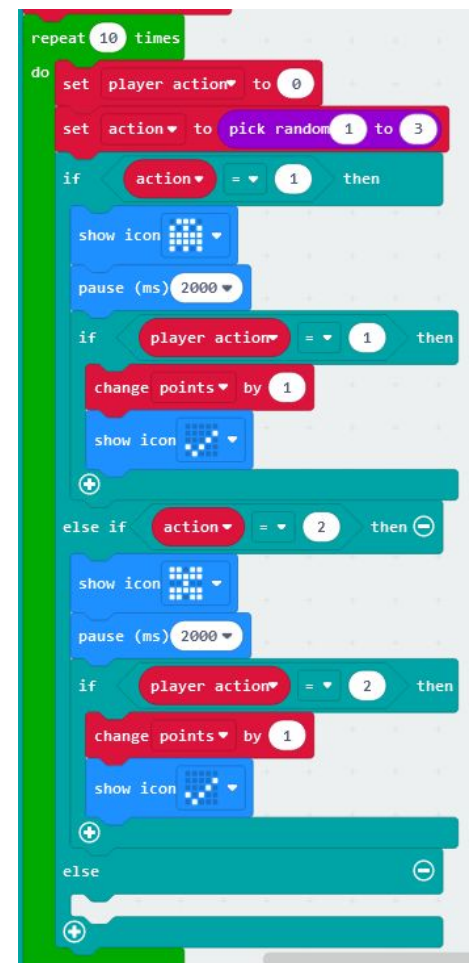
- 15) I found one other “glitch” with our code at this stage. If you shake the micro:bit right at the beginning of the program, it will set the “player action” variable to 1. We never reset the “player action” variable in the code, so if the first 3 actions are all ghosts, the game will give us three points, even if we only shake for the first ghost. To fix this, let’s set player action to 0 at the top of our green repeat loop.



Take Home

For the take home, we will be adding another action and response.

- 1) Let’s add a third action. First, set action to pick a random number from 1 to 3. Find the very bottom of the “if action = 1” frame, and click the plus sign there, to add another else if conditional. Drag the “show icon *butterfly*” block up into this new empty space. We chose the butterfly as action number 2, so the condition for this block will be “action = 2”. The inside of your green loop should now look like this



Micro:Bits Bop-It!



- 1) Let's fill out the empty "else" frame by duplicating the pattern from the butterfly and the ghost. We will choose a new icon, check if player action = 3, then change points if correct. My new icon will be the umbrella. The inside of the else block will look like this:

- 2) Now, we need to add to the forever loop so that it checks for our third action. I am going to duplicate the "if is shake gesture" frame, then change "shake" to "logo down" and change player action to be 3. That forever frame now looks like this:

- 3) Now we have four actions! You might find that it is hard for the Micro:bit to tell the "shake" and the "logo down" gestures apart. If that is the case, you might want to change them to be "tilt left" and "tilt right". There are a lot of Micro:bit inputs to explore!

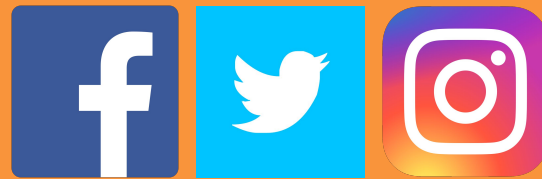
Challenge Questions:

- How could you add a high score feature to this program?
- How could you add to this program so that it displays an X when you don't do the correct action in time?
- How could you make the rounds speed up as you score more points?

#SVatHome

Want to share your
project or results with us?

Email or tag us
@ScienceVenture



Have a question?

Reach us at
svcamp@engr.uvic.ca