Recipe 6: Maze Runner

By the end of this activity, you will have a game that requires the player to tilt the micro:bit to guide their flashing dot through a maze drawn on the LED screen.

Section 1

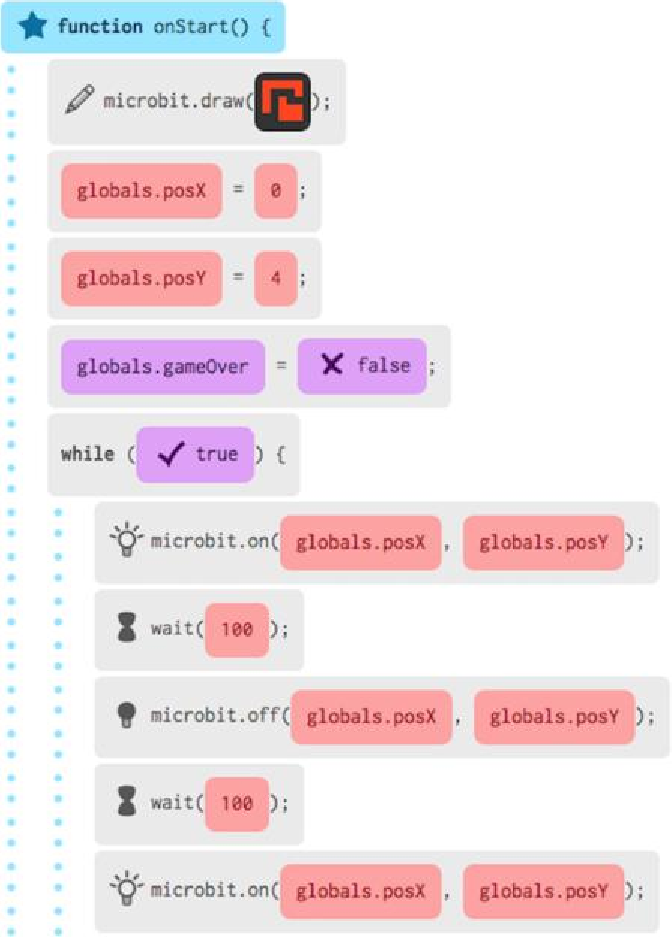
1. Draw the maze you want players to solve - be sure there is a place to start and a place to finish.

2. Store the player’s current position in two variables, called posX and posY.

3. Set them to be wherever the player’s dot should start.

4. The game will only end when the puzzle is solved, so use a while-true loop to keep the game running forever.

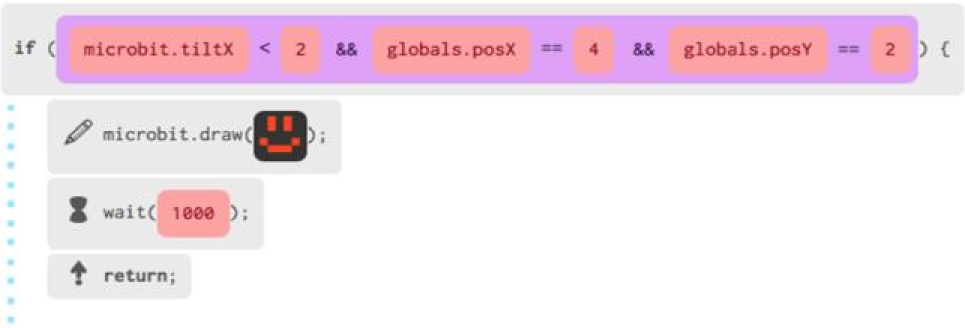
5. We need a way to differentiate between the maze and the player’s dot (given by the posX and posY variables). One way to do this is to have the player’s dot flashing, by turning it off briefly and then on.



```
function onStart() {
    microbit.draw(▣);
    globals.posX = 0;
    globals.posY = 4;
    globals.gameOver = ✕ false;
    while ( ✓ true ) {
        microbit.on( globals.posX , globals.posY );
        wait( 100 );
        microbit.off( globals.posX , globals.posY );
        wait( 100 );
        microbit.on( globals.posX , globals.posY );
```

Section 2

1. If the micro:bit is being tilted in the right direction, and the player’s dot is at the end of the maze, then the maze puzzle has been successfully completed. Draw a happy face and then end the program!



```
if ( microbit.tiltX < 2 && globals.posX == 4 && globals.posY == 2 ) {
    microbit.draw( );
    wait( 1000 );
    return;
```

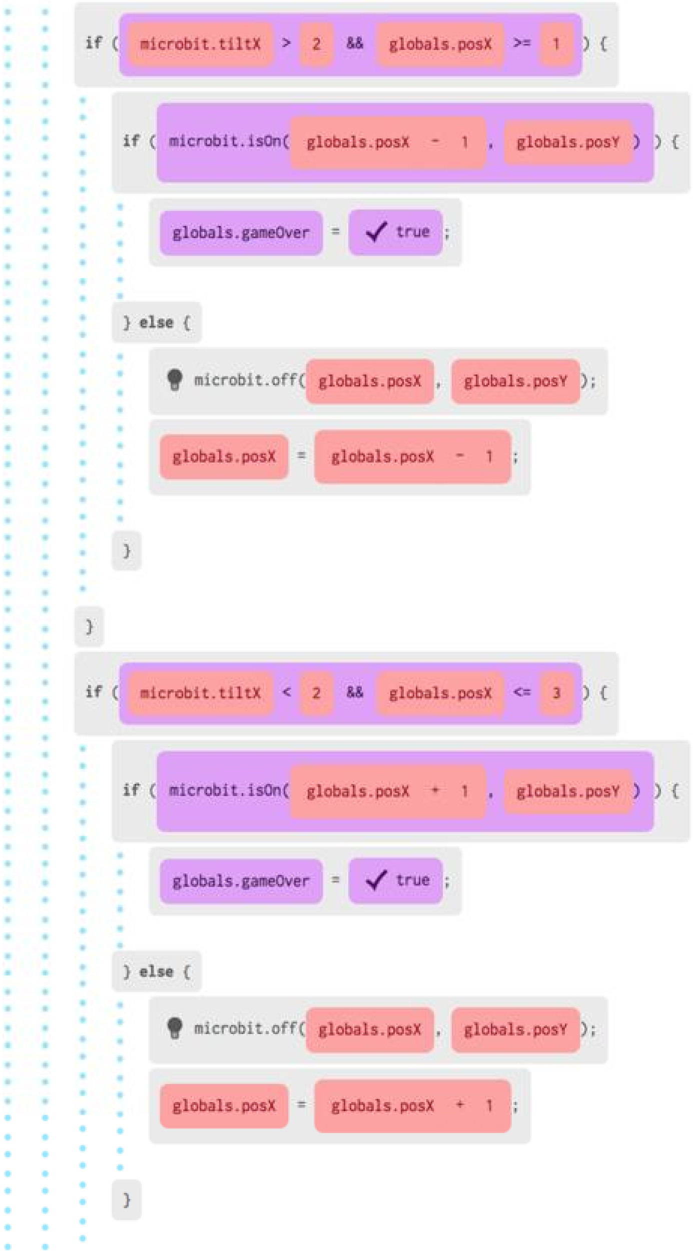
Section 3

1. If the micro:bit is being tilted to the left (and it’s not on the edge), check that the LED at the point the player dot wants to move to is not on.

2. If it is, the player has hit a wall and the game is over (so turn the gameOver variable to true). Otherwise, update the player’s dot position (given by posX and posY) to be one to the left of where it was. Turn the LED off at the player’s old dot position.

3. If the micro:bit is being tilted to the right (and it’s not on the edge), check that the LED at the point the player dot wants to move to is not on.

4. If it is, the player has hit a wall and the game is over (so turn the gameOver variable to true). Otherwise, update the player’s dot position (given by posX and posY) to be one to the right of where it was. Turn the LED off at the player’s old dot position.



```
if ( microbit.tiltX > 2 && globals.posX >= 1 ) {
    if ( microbit.isOn( globals.posX - 1 , globals.posY ) ) {
        globals.gameOver = ✓ true ;
    } else {
        microbit.off( globals.posX , globals.posY );
        globals.posX = globals.posX - 1 ;
    }
}
if ( microbit.tiltX < 2 && globals.posX <= 3 ) {
    if ( microbit.isOn( globals.posX + 1 , globals.posY ) ) {
        globals.gameOver = ✓ true ;
    } else {
        microbit.off( globals.posX , globals.posY );
        globals.posX = globals.posX + 1 ;
    }
```

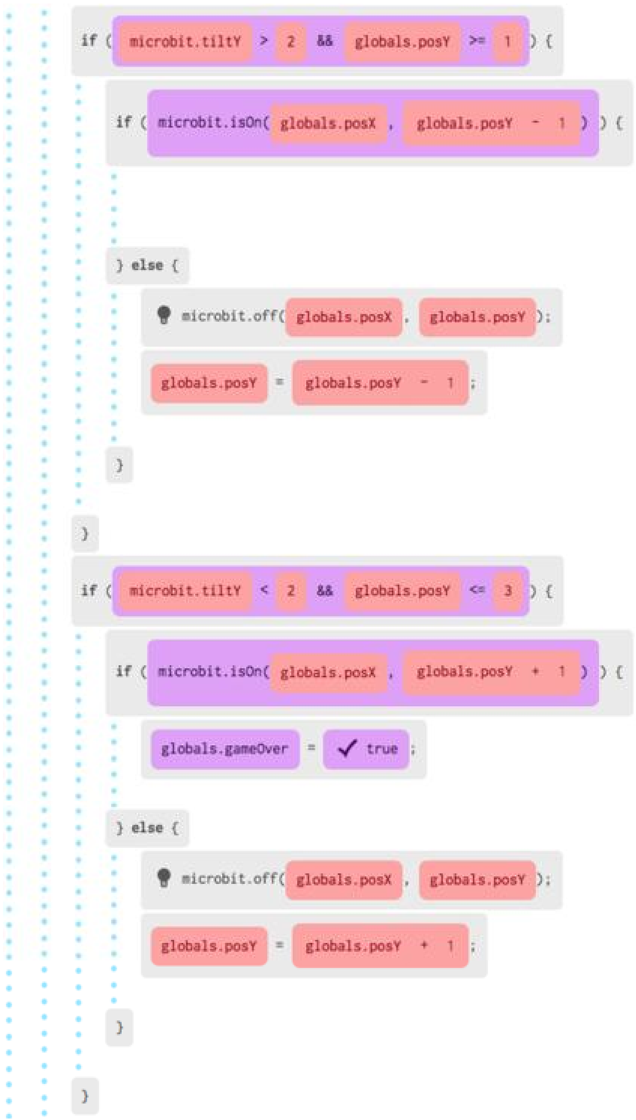
Section 4

1. If the micro:bit is being tilted up (and it’s not at the top), check that the LED at the point the player dot wants to move to is not on.

2. If it is, the player has hit a wall and the game is over (so turn the gameOver variable to true). Otherwise update the player’s dot position (given by posX and posY) to be one above where it was. Turn the LED off at the player’s old dot position.

3. If the micro:bit is being tilted down (and it’s not at the bottom), check that the LED at the point the player dot wants to move to is not on.

4. If it is, the player has hit a wall and the game is over (so turn the gameOver variable to true). Otherwise, update the player’s dot position (given by posX and posY) to be one below where it was. Turn the LED off at the player’s old dot position.

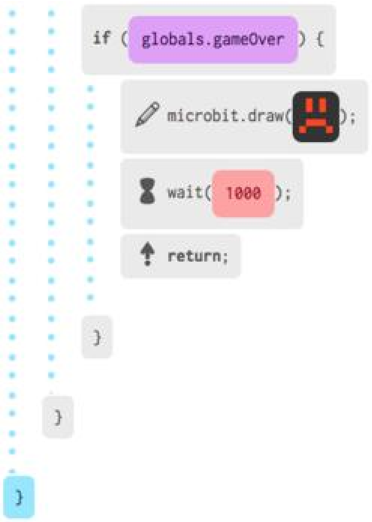


```
if ( microbit.tiltY > 2 && globals.posY >= 1 ) {
    if ( microbit.isOn( globals.posX , globals.posY - 1 ) ) {

    } else {
        microbit.off( globals.posX , globals.posY );
        globals.posY = globals.posY - 1 ;
    }
}
if ( microbit.tiltY < 2 && globals.posY <= 3 ) {
    if ( microbit.isOn( globals.posX , globals.posY + 1 ) ) {
        globals.gameOver = ✓ true ;
    } else {
        microbit.off( globals.posX , globals.posY );
        globals.posY = globals.posY + 1 ;
    }
}
```

Section 5

1. If, after all these checks about the tilting, the gameOver variable was set to true (because the player hit a wall), draw a sad face to let the player know and exit the program.



```
if ( globals.gameOver ) {
    microbit.draw( );
    wait( 1000 );
    return;
}
}
}
```