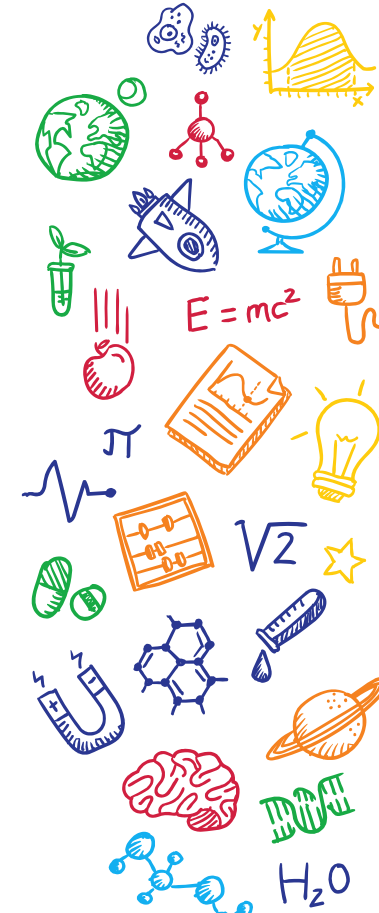
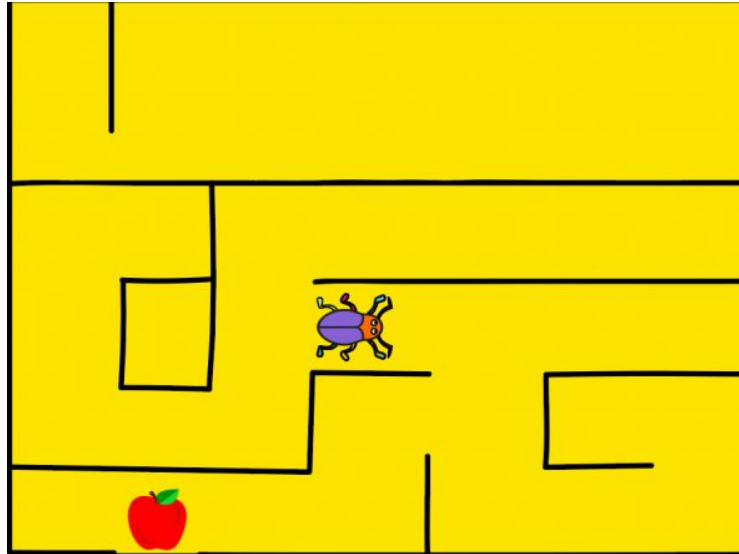


Beetle in the Maze



Beetle in the Maze

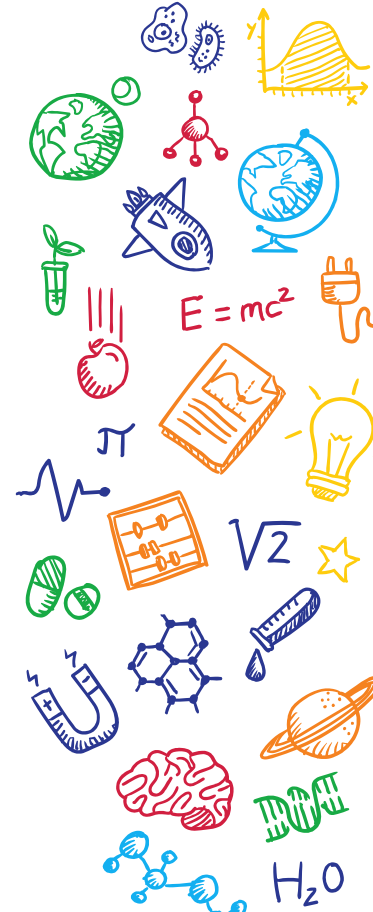
- In this activity you will make an animated game in PictoBlox where beetle needs help to come out of the maze without touching the boundaries.



Setting up the Stage

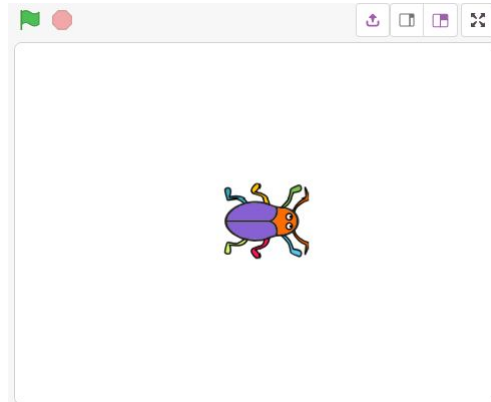
Setting up the Stage

- Right below the stage, you will see a purple button with a bear on it. Take the cursor over it and select **Choose a Sprite**; the sprite library will open.
- Select the beetle sprite from the library.



Setting up the Stage

- Now, there are two sprites: beetle and Tobi.
- Right click on the icon for Tobi (below the stage) and click on the delete option.



Sprite: Tobi x 0 y 0

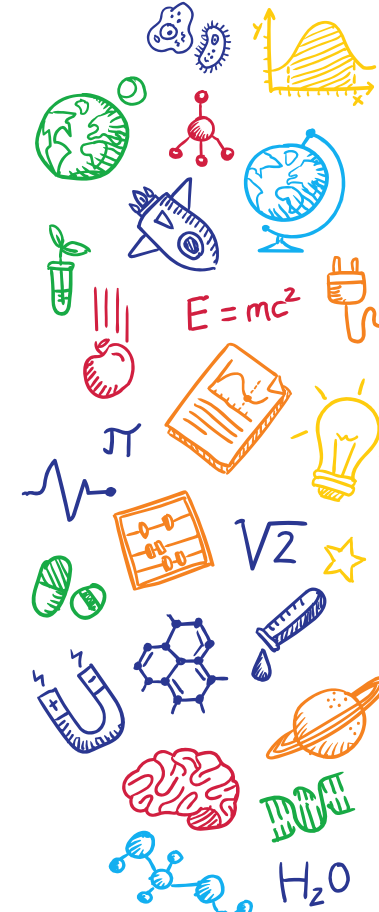
Size: 100 Direction: 90

Stage

Backdrops: 1

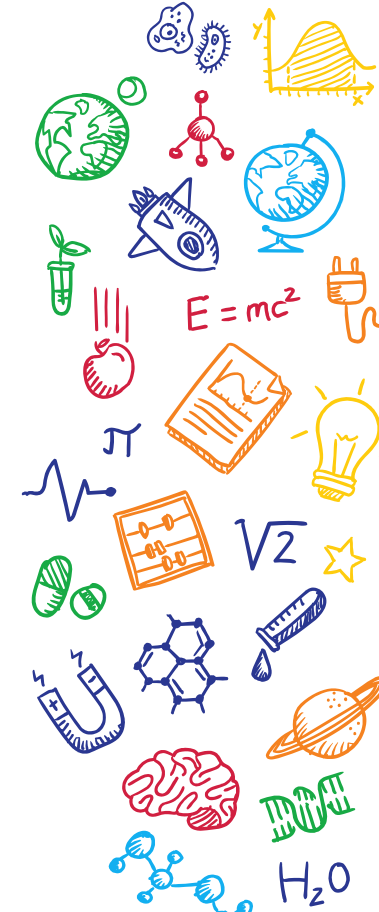
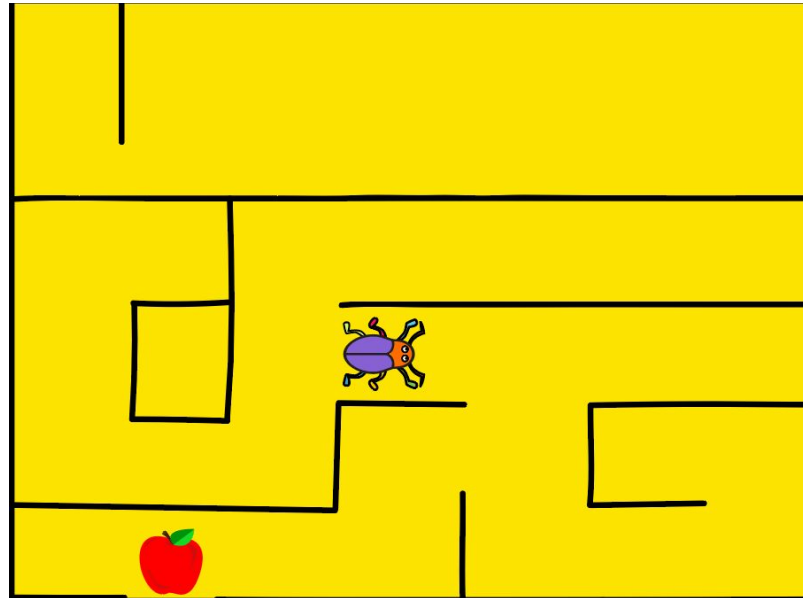
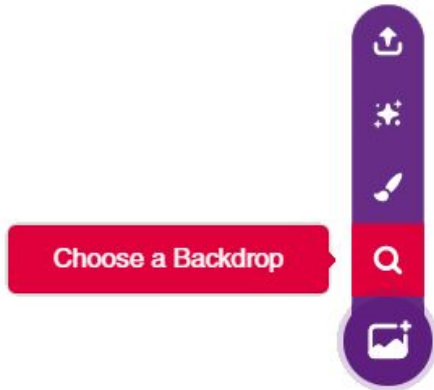
Sprite list: Tobi, Beetle

Context menu for Tobi: duplicate, delete, export



Setting up the Stage

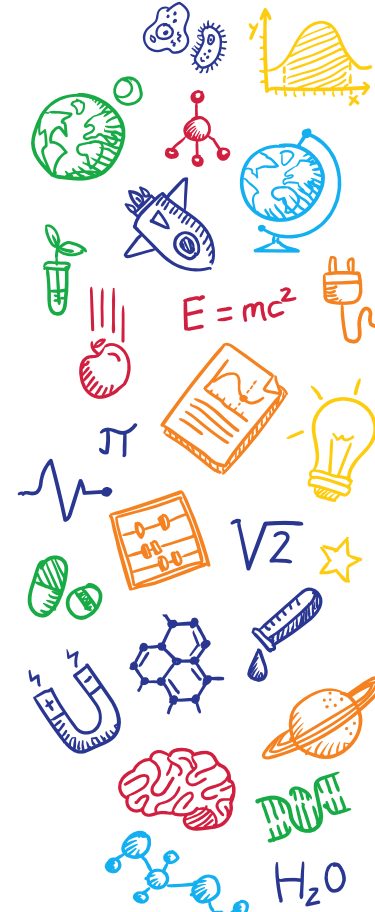
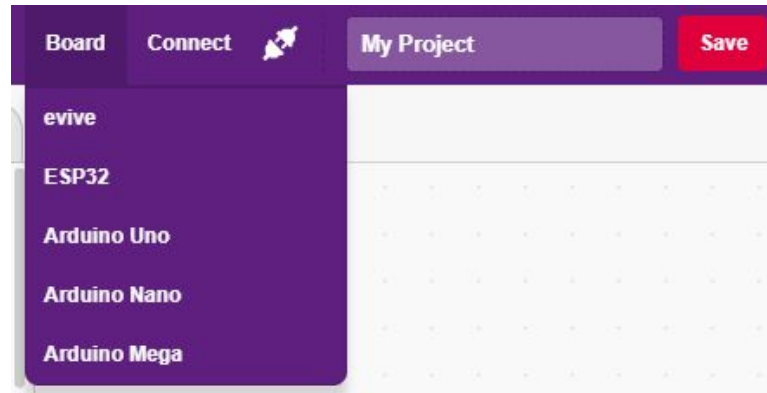
- Click on the purple colored icon which says **Choose a Backdrop**.
- Select the maze backdrop from the library.



Giving the Beetle Its Moves

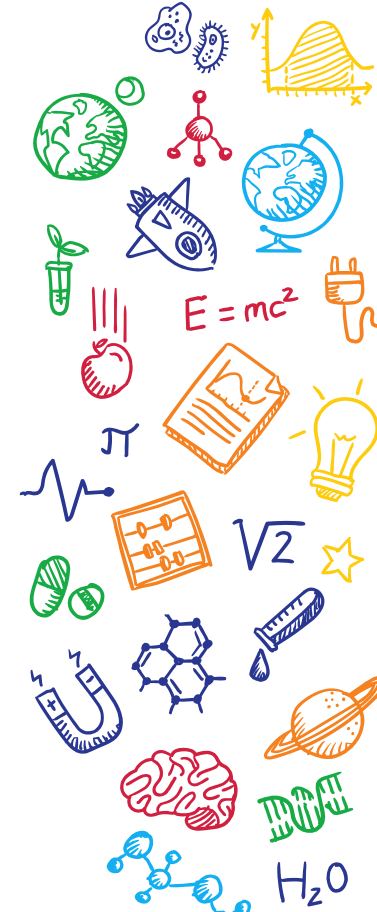
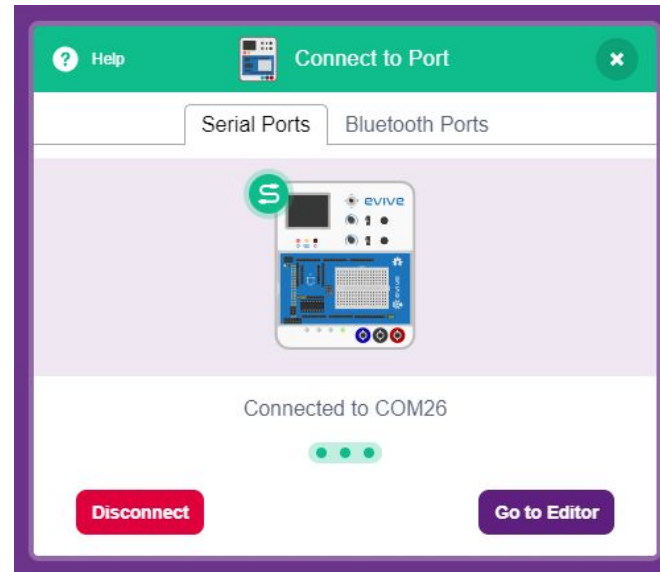
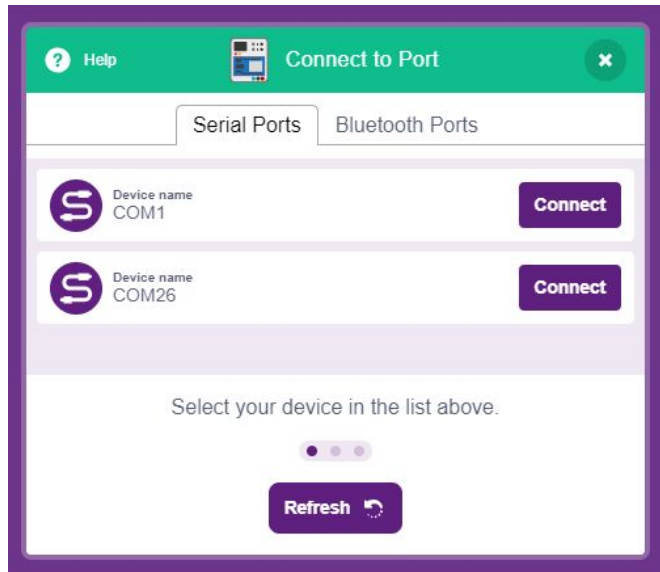
Interfacing evive with PictoBlox

- Connect evive to your laptop/PC and open PictoBlox.
- In PictoBlox, go to the menu and click on the Boards
Select the evive.



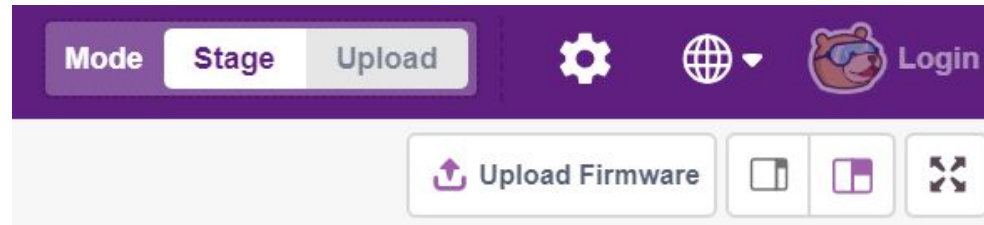
Interfacing evive with PictoBlox

- Once you've selected the board, click on the Connect tab and connect the board.

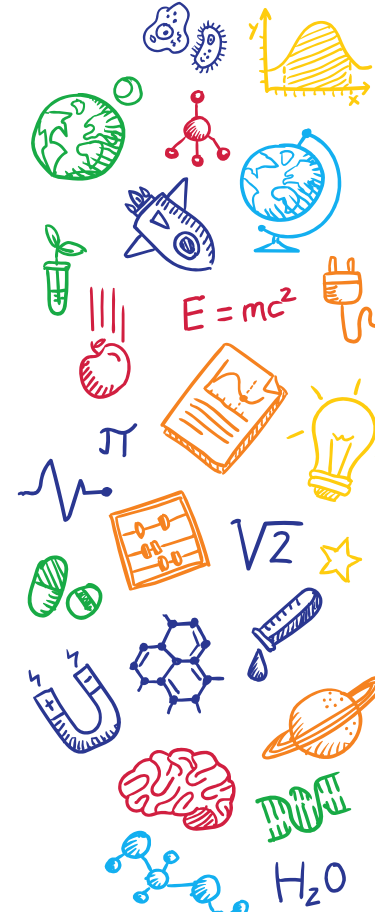


Upload Firmware


- You have to upload the firmware into the device before interacting with it.
- Click on **Upload Firmware** button.

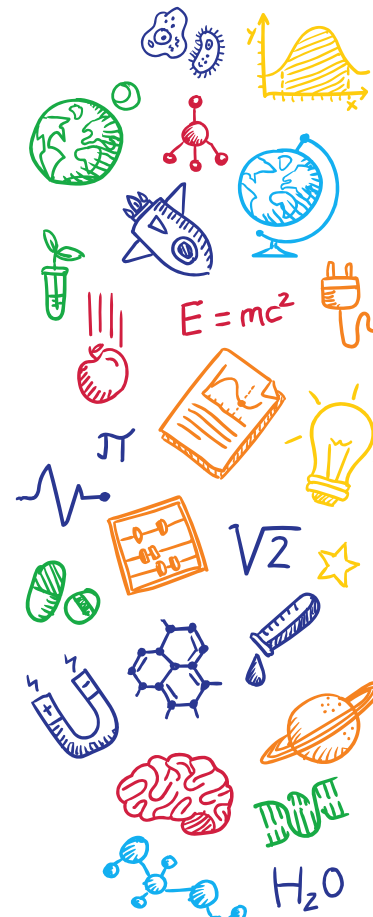
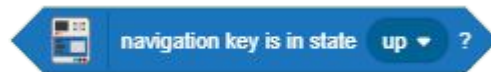


- Once you successfully uploaded the firmware you will see evive logo on evive display.



Giving the Beetle Its Moves

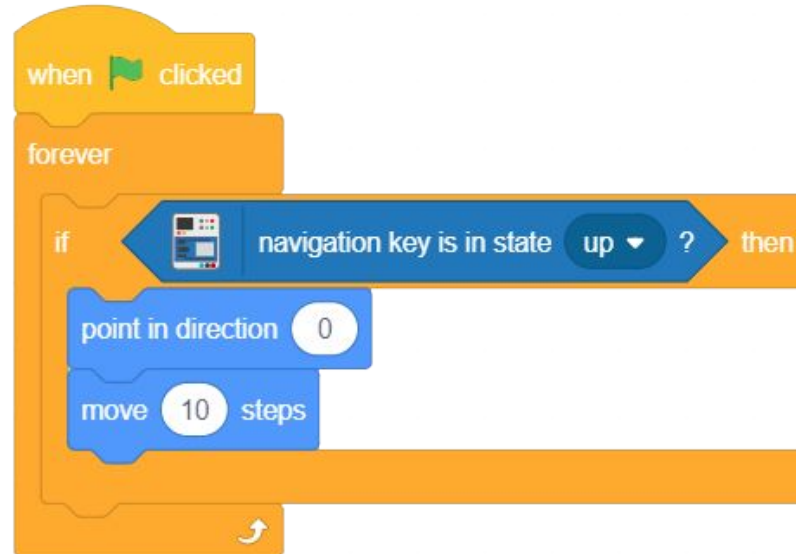
- We will control the Beetle movement using Navigation Key of evive.
- The beetle must move in the direction we push the key.
- **navigation key is in state**  ? block checks if the navigation key is in one of the four particular directions. The direction for which you wish to check is your choice and can be chosen from the drop down on the block.



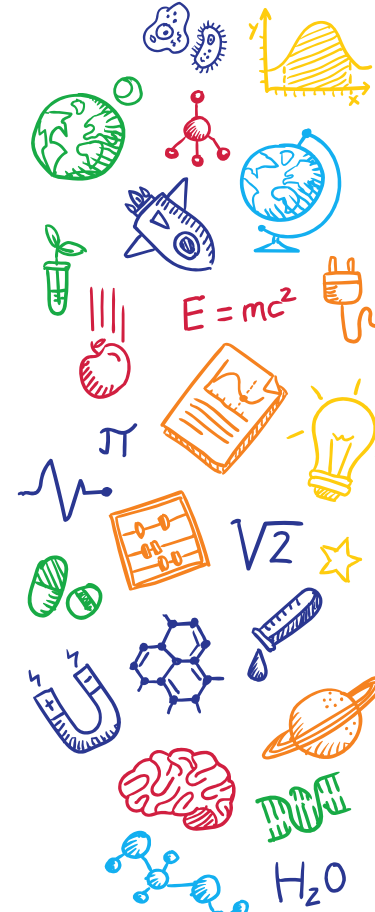
Giving the Beetle Its Moves

- Write the code to move the beetle in up direction & test:

```
when clicked
  forever
    if navigation key is in state up ? then
      point in direction 0
      move 10 steps
```

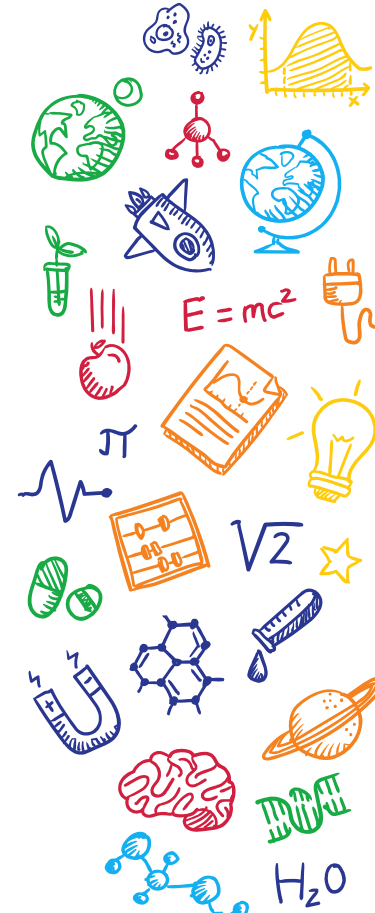


- Try to make the code for other directions.



Giving the Beetle Its Moves

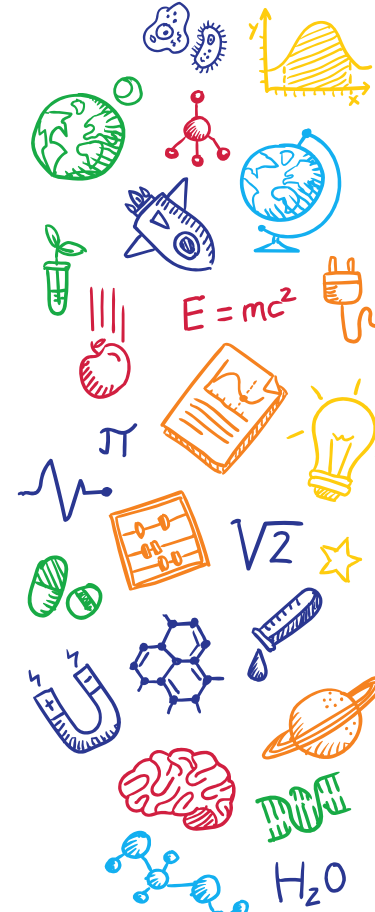
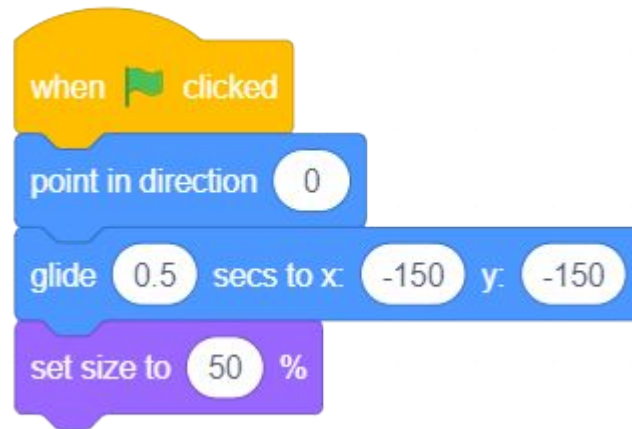
- For 'down', set the direction as 180.
- For 'right', set the direction as 90.
- For 'left', set the direction as -90.



Sensing the environment

Sensing the environment

- Now, it's time to write the script for safely taking the beetle to the end of the maze.
- Make the code to initialize the game, so that every time game starts at one position.

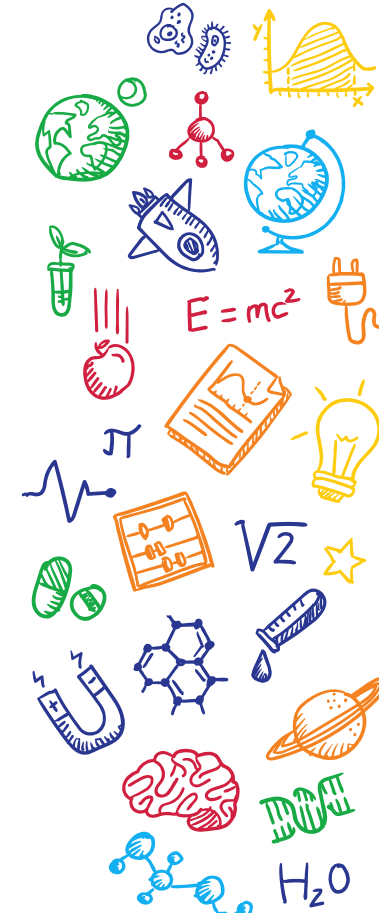


Sensing the environment

- Now we will add the condition when the beetle touch maze, it will go back to its original position.
- Using the color picker, pick the maze color.

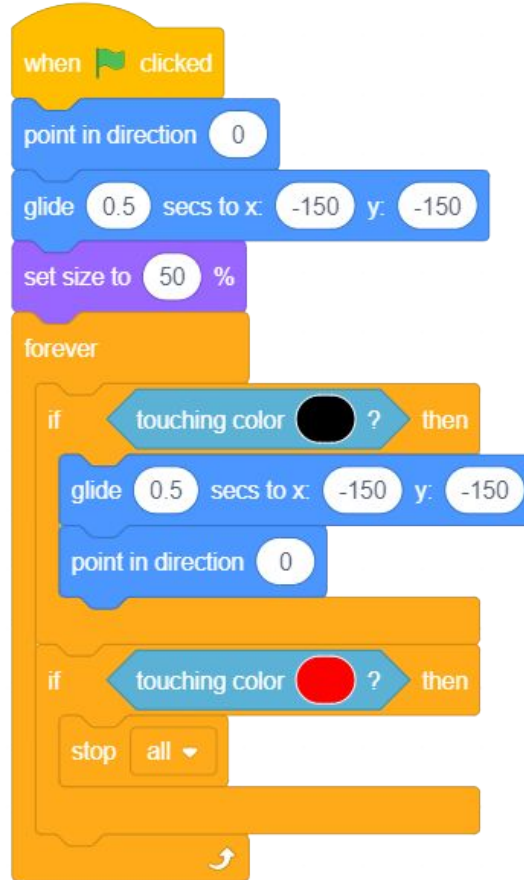
```

when clicked
  point in direction 0
  glide 0.5 secs to x: -150 y: -150
  set size to 50 %
  forever
    if touching color [black] ? then
      glide 0.5 secs to x: -150 y: -150
      point in direction 0
  
```

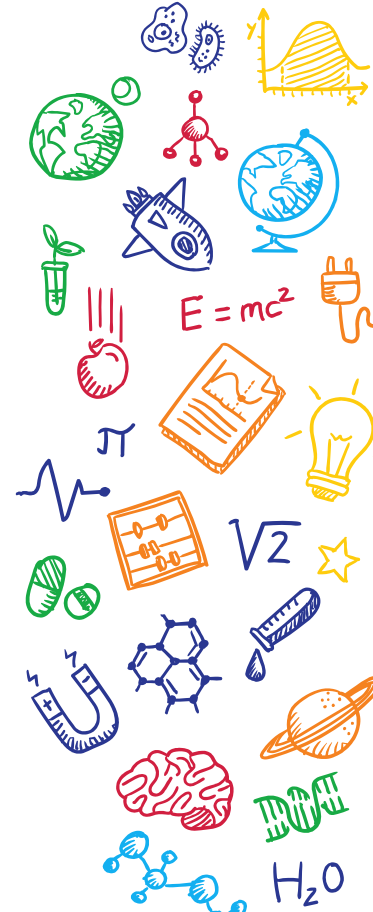


Sensing the environment

- Now we will add the condition to end the game. Game ends when the beetle touches the apple.
- Using the color picker, pick the apple color.
- Game is complete now. Play!



```
when clicked
point in direction 0
glide 0.5 secs to x: -150 y: -150
set size to 50 %
forever
if touching color black ? then
glide 0.5 secs to x: -150 y: -150
point in direction 0
if touching color red ? then
stop all
```



THANK
YOU

