

# Snap! Looping and costumes

COR100 Snap! Programming Fall 2022



## Green flag

- The project GreenFlag contains two sprites. Each sprite has its own script. There is no way to run both these scripts simultaneously.



Figure 2: two scripts for two sprites on one stage

- When attaching the *green flag* block to each of the scripts, they will start at the same time.

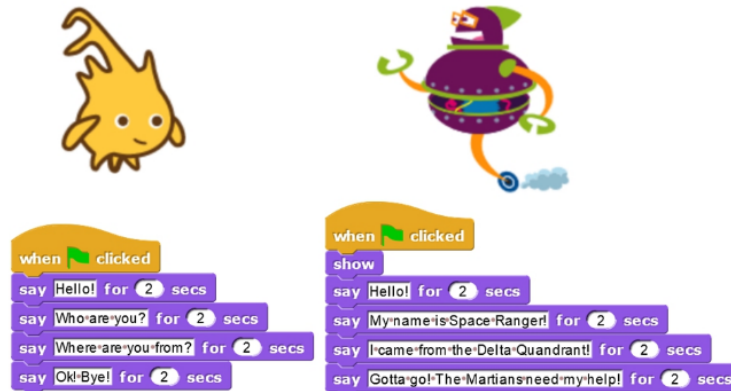


Figure 3: two scripts for two sprites on one stage with green flag

- But this is a conversation: to build in pauses, use the wait N secs command:



Figure 4: a conversation between two characters

- See project [GreenFlag](#) in the cloud ([video/GDrive](#))

## Simple Looping



- See project [Looping](#) online

## Looping commands

- Looping (or *iteration*) is the repetition of a sequence of commands
- The commands repeat and forever in Snap! allow simple looping
- [ ]

What does this command do?



Figure 6: simple loop with repeat

The sprite will move around in a square. Each move takes it 100 steps before it turns 90 degrees.

- [ ]

What does this command do?



Figure 7: simple loop with forever

The sprite will spin around itself forever - one full spin will take  $360/5 = 72$  iterations.

## Jumping up and down

- To make a sprite jump up and down repeatedly:
  1. get the basic command sequence
  2. repeat the sequence
- This command moves the sprite along the y-axis (vertically):



- This command brings it back to the origin:



- Finally, we use repeat to iterate four times:



- [See the code here.](#) The XY-geometry of the background will be covered in an upcoming lesson.

## Smooth motion

- Place your sprite somewhere near the left edge, add and click the following script:

#caption: seemingly instantaneous jump to the right



- Looping reveals that these commands are not instantaneous: the sprite moves the same number of 300 =  $10 * 30$  steps.



#caption: loop over a move to the right

- Reducing the number of steps per iteration and increasing the number of iterations still moves 300 =  $30 * 10$  steps, but the movement is now much smoother.



#caption: loop over a move to the right

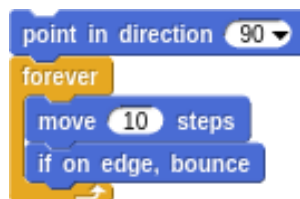
## Practice - looping

1. Define a new project called "Looping".
2. Create three sprites and pick different costumes for them.
3. Make the first sprite spin around itself really slowly forever. Tip: slow motion means many small degree changes.
4. Make the second sprite jump up and down 10 times: the sprite should move down slowly (e.g. 2 seconds) and bounce up fast (e.g. 0.5 seconds).
5. Make the third sprite move smoothly 200 steps to the right. Tip: to bring a sprite back that has left the stage, move it with a negative number of steps - e.g. -200.

[Practice solution - looping](#)

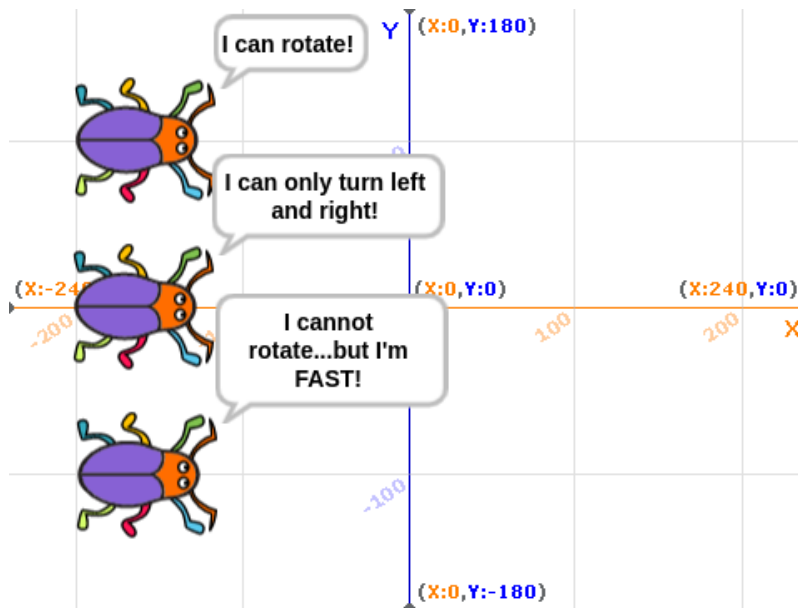
## Rotation style

- If you want your sprite to walk horizontally (East-West) forever, let it bounce upon hitting an edge.



- You can control the orientation of the sprite after bouncing using the rotation style buttons located left of the little sprite image that indicates the currently chosen rotation style.

## Practice rotation styles

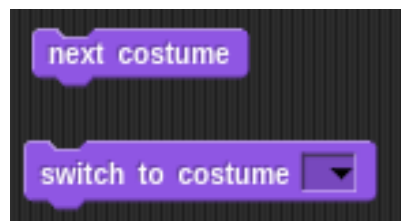


1. Define a project that implements the three rotation styles as shown [in this video \(GDrive\)](#).
2. Start by defining three sprites, and download the beetle costume. Tip: you can download it once and drag it onto a sprite in the sprite area below the stage.
3. Script: point the beetle in the same direction (90). In a forever loop, make it move 10 steps and if on edge, bounce it.
4. Choose a different rotation style for each beetle.
5. You need three identical scripts (remember a script is tied to a sprite). To start all three beetles at the same time, add a Green Flag start command.
6. *Optional:* once you're done, add speech and experiment with making them start at different times, move at different paces, and add the XY geometry background.

[Practice solution - rotation style](#)

## Animation using costumes

- Animation (images brought to life) is an illusion of action or motion, a trick played on our eyes and ears
- To create this illusion use series of sprite costumes:



## Practice animation using costumes

1. Define a new project and call it "animation".
2. Make a sprite and name it "walk".
3. Open the *Costumes* menu of the sprite. Download 4 costumes of "*avery walking ...*" from the library.
4. Show avery dragging her feet: write a script that makes her move to the right 10 steps at a time whenever the space bar is pressed. Do this a few times, then drag avery back to her starting position.
5. Add the command "next costume" at the end of the script and run it again: avery now seems to walk to the right side of the stage. In fact, you move through four different static costumes.
6. Make avery walk *frantically* off stage: enclose the last script in a "forever" loop and add a green flag starting command at the top.



7. To bring the sprite back to the stage, right click in the stage area and choose "Show all". You'll have to drag the sprite to the starting position.
8. To stop the frantic motion, add a "wait 0.2 secs" command at the end of the script. Avery now walks normally.
9. [ ]

How could you stop avery from walking off stage?

Answer:

Add "if on edge, bounce" after the "wait" command inside the "forever" loop, and change the rotation style to "only face left/right":



[Practice solution: Project animation](#)

Created: 2022-10-11 Tue 07:02