

# Snap! Sequence and sounds

UBMS Game and Robo Programming with Snap! and Python

June 27, 2023



## Sequences

- The word *sequence* comes from the Latin word *sequi*, "to follow".
- Computers require *exact* sequences of steps or statements to work - like a *to do* list:
  1. Rise and wake puppy.
  2. Feed puppy.
  3. Walk puppy.
- *Deviations* from the prescribed sequence lead to errors.
- The worst errors are those that remain *undiscovered*: the script will run but it won't do what we expect it to do, and *debugging* is hard.
- A script like ?? below represents a series of steps.

## Practice - Sequences

1. Define a new project "Sequence" and build this script in your Snap! dashboard. You can pick a non-standard sprite if you like.

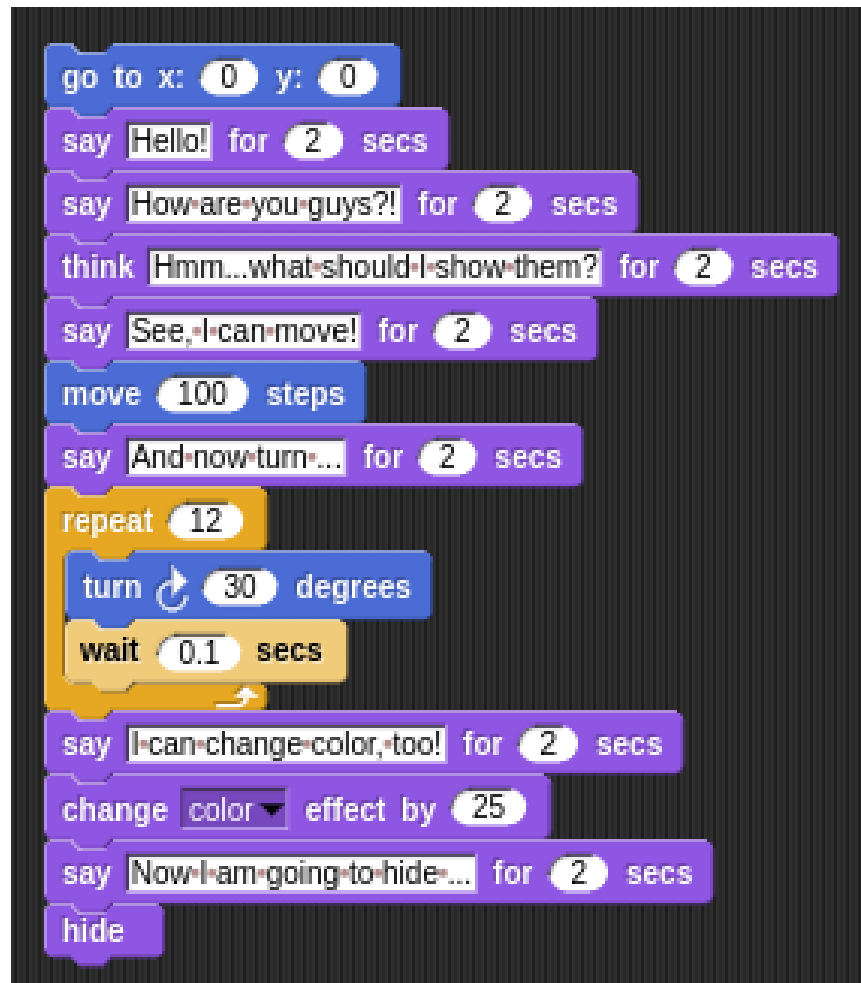


Figure 1: sample Snap! script (Source: Joshi)

2. When you try to run the script a second time, nothing happens. What could you do to re-run it properly? - *Tip: check "Looks" commands.* Add the `Looks::show` command at the top.
3. This is how the result should look like (video).



Figure 2: screenshot - sample Snap! script (Source: Joshi)

4. Make the change and re-run the script.
5. Save the Sequence project to the cloud and find it in **My Projects**.

## Sounds

- Every sprite has a **Sounds** tab
- You can *import* an existing sound from the Snap! library (**Sounds** option in the main menu), or you can *upload* your own sound (mp3).
- The **play sound ... until done** command moves to the next statement only if the sound file has been played.
- The **play sound...** command starts playing the sound file and moves immediately on to the next statement.
- You can stop all sounds with the **stop all sounds** command.

☐ What does this script sound like?



Figure 3: dog sprite with library barking sound



Figure 4: Play sound ... until done



Figure 5: Play sound ...



Figure 6: Play sound ...

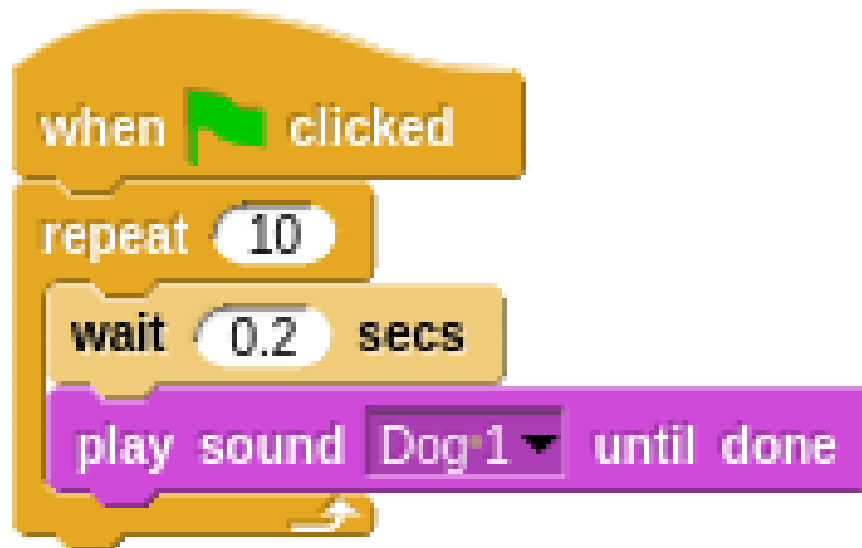


Figure 7: Script with sounds

- What does this script sound like?

### Practice (in class) - Sounds

1. Define a new project "SoundCheck".
2. Pick a costume and a suitable sound.
3. Write a script that produces 5 consecutive sounds **without loop**.
4. **Duplicate** the script and add a **repeat** loop with only one sound command
5. Save the project to your My Projects list

Practice solution - soundcheck

### Program 2 (in class)

- In the remaining time, solve the 2nd programming problem ("Sound-bites" in Canvas)



Figure 8: Script with sounds

- If you cannot finish it in class, finish it at home or during study time (by tonight at 10 pm)
- We will look at the solution tomorrow and also have a 3rd quiz!

### Bonus problem

- When clicked, your program should play a composition (sequence of sounds). A simple example from a previous course is [here](#). Here is another one with a choice of music (much more complicated code).
- You must however take care to NOT use music that has a COPYRIGHT on it: either compose yourself, or use free music (e.g. Wikimedia)

### Summary

- Computers required exact syntactically correct sequences of instructions to work at all.

- A typical animation sequence includes at least code from: MOTION (sprite movement), CONTROL (program flow), and LOOKS (appearance and messages).
- You can upload sounds from the web (as long as the copyright permits it), or from the Snap! cloud platform, or record your own directly in the editor.
- Some commands will have to complete an action until the script can continue, others will cause an action and simply continue even if the action is not completed.