

Game Programming with Snap!

STEMonstration, Lyon College Academic Day

Marcus Birkenkrahe

March 4, 2023

Introduction



View this presentation any time at: bit.ly/3mfNEaT

Is this your first time?



- Has any of you already programmed?
- Have you ever created an animation or a game?
- Have you created a web page or used a "high level" language like Python, C, C++, Java or JavaScript?
- How would you describe the experience of programming?

Session overview



- Introduction (15 min)
- Getting started with Snap! (30 min)
- Building a simple animation (30 min)
- Building a simple game (30 min)
- Summary and wrap-up (15 min)
- **Why am I teaching this course?** Gamer, programmer, teacher.
- **Background:** teaching data science and computer science
- **Funny accent:** I am German, came to Batesville 2 years ago

Why Snap!?



["Donkey Kong" - Ty Ferguson, Clay](#)



["Pacman" - Matthew Wisdom, Tyler Landry, Rylan Turks](#)

"Snap! Oh Snap!" Lyon College Course - Fall'22

- **COR100/Y1:** Lyon's 1hr/week course teasers, fall'23 Snap! 12 students, weekly meetings of 50 minutes learning Snap! together.
- Final team projects: Retro Arcade Games - Donkey Kong, Pacman, Space Invaders, and Asteroids.
- Compared to other programming languages, visual, block-based languages are very forgiving (computers are extremely picky and panic easily) - attention to detail is key

How I got started



Open project: bit.ly/SnapTimeMachine

What is Snap?

The screenshot shows the Snap! website at snap.berkeley.edu. The interface features a top navigation bar with links for Run Snap!, Explore, Forum, Join, Log In, and Help. Below the navigation is a search bar and a user profile icon. The main area displays a block-based programming environment. On the left, a script is shown with various blocks: 'pipe' (with 'The Software For Novice And Advanced Programmers' comment), 'split [by whitespace]', 'keep items such that [length of [] > 3 from []]', 'map [letter 1 of [] over []]', and 'combine with [join [] items of []]'. To the right, there is a large orange block labeled 'no ceiling' with the text 'SNAP!' above it. Below this are several other blocks, including 'pipe []', 'if [is [pipe] empty?]', 'report [value]', and 'pipe [call [item 1 of [pipe] with inputs: value] report [input list: [all but first of [pipe]]]]. At the bottom of the page, there is a 'Welcome to Snap!' message and a brief description: 'Snap! is a broadly inviting programming language for kids and adults that's also a platform for serious study of computer science.' Below this are buttons for 'Run Snap! Now', 'Example Projects', and 'Reference Manual'.

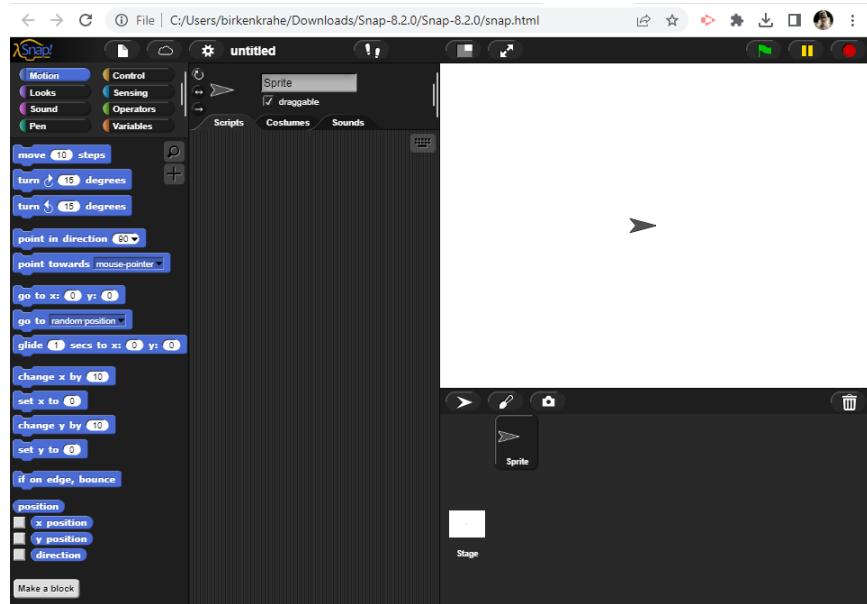
- Web-based HTML5 online application
- Block-based visual programming language

- Create animations, games, simulations
- Open source and free to use
- Developed at U. Berkeley, bought by SAP
- <Right-Mouse> View page source (HTML)
- <Right-Mouse> Inspect (HTML, CSS, JavaScript, Media)
- Developed by German programmer Jens Moeinig
- SAP is Germany's largest software company (ERP systems)
- Unlike MIT's Scratch, Snap! can do anything
- In programming, everything depends on data structures, e.g. numbers, strings (that make up texts), tables, lists/dictionaries etc.
- Languages "for anything" allow you to build such structures

Getting started with Snap!



Download the desktop app



- In your browser, go to bit.ly/3kTxX8O
- Download the .zip file
- Extract the file to **Downloads**
- Open **Downloads/Snap-8.2.0/snap.html**

Run Snap! in the cloud

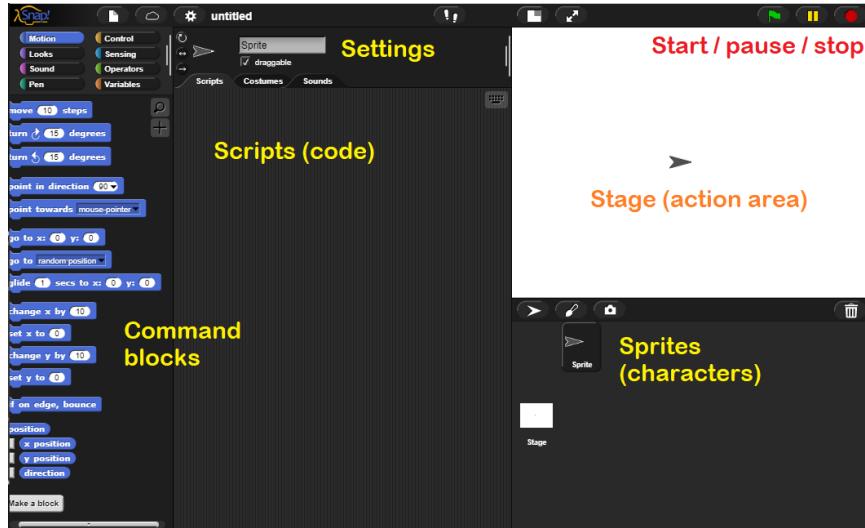
The screenshot shows the homepage of snap.berkeley.edu. At the top, there are navigation links for "Run Snap!", "Explore", and "Forum". A search bar is also present. On the left, there's a yellow cartoon character and some sample code blocks: "when green flag clicked", "clear", "pen up", "go to x: 0 y: -150", "pen down", "point in direction 0", and "vee". To the right of the code is a drawing of a fractal tree. The right side of the page features a sidebar with a dropdown menu for the user "birkenkrahe". The menu includes options like "My Projects", "My Collections", "My Public Page", "Followed Projects", "My Profile", and "Log Out". Below the sidebar, there are buttons for "Run Snap! Now", "My Projects", "My Public Page", "Example Projects", and "Reference Manual".

Welcome, birkenkrahe!

Snap! is a broadly inviting programming language for kids and adults that's also a platform for serious study of computer science.

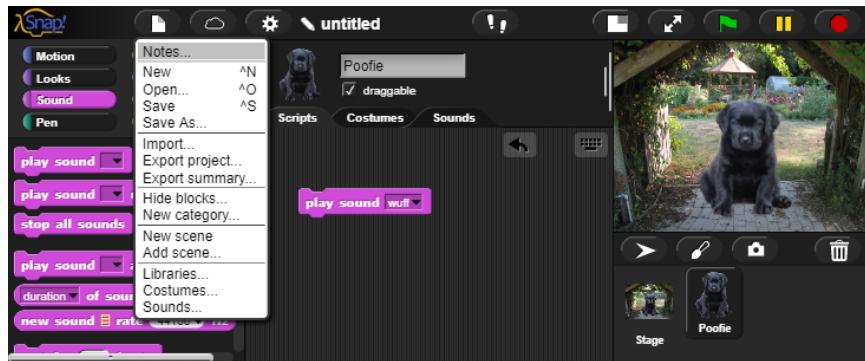
- Go to snap.berkeley.edu
- Click on Run Snap! to program
- Click on Join to keep your projects
- Pick a simple password
- Check your email for verification

Snap! User Interface (UI)



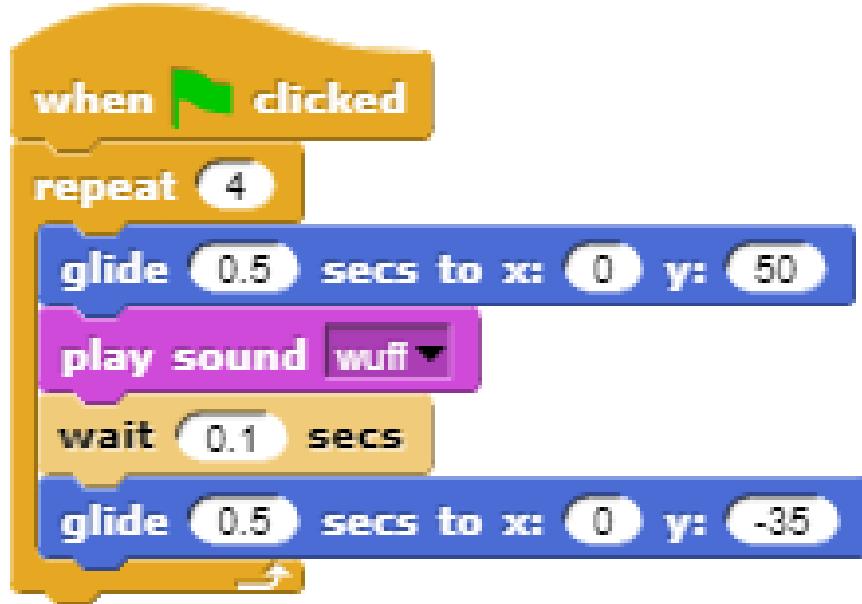
- Sprites are controlled by scripts
- Scripts consist of command blocks
- Sprites (characters) act on stage
- Settings add control, input and output
- Drag the "turtle" sprite around a little
- Drag a "move 10 steps" block to the script area
- Click the block / change 10 to 100 and click again

Sprites, costumes, sounds

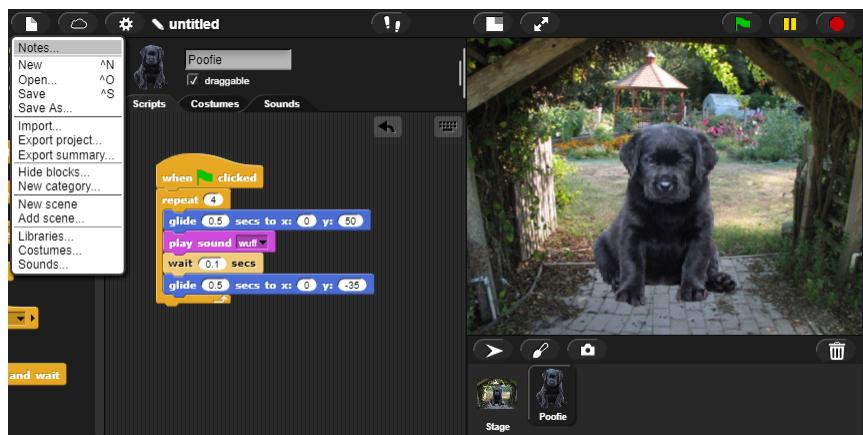


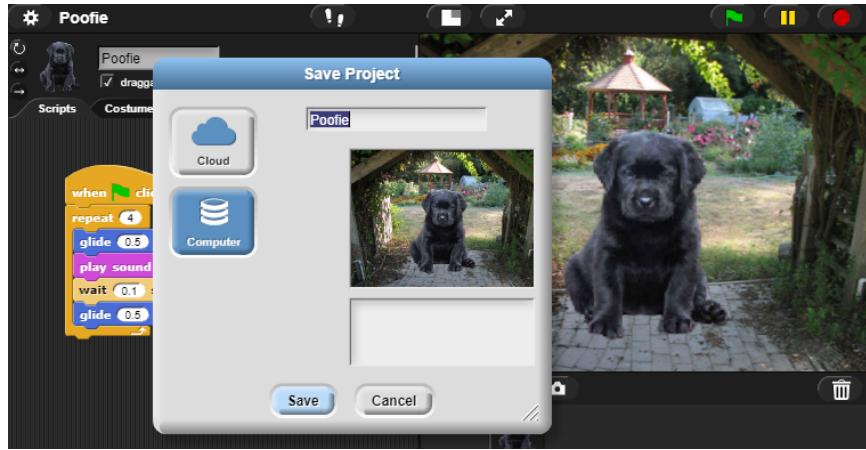
- New **sprites** are created as "turtles"
- You can create sprite **costumes** and **sounds**
- Every sprite is controlled by its **script**
- Create your own barking Poofie now!
- Click on the sprite symbol below the stage
- Open the **Costumes** tab next to the Scripts tab
- Open the file menu at the top
- Open the **Costumes** menu in the file menu
- Find the **dog** costume (or **cat** if you prefer)
- Click on the costume and **Import** it
- Click on the **Sounds** tab
- Open the **Sounds** menu, find **Dog 1** and **Import** it
- Rename the sound **wuff**
- Rename the Sprite **Poofie**
- Open the **Scripts** tab
- Open the **Sound** command block section
- Drag **play sound** into the script area
- Select **wuff** and click the command block

Write your first script



Save your first project





If you're registered

Snap!

Run Snap! Explore Forum

birkenkrahe ..

Poofie by birkenkrahe

"Poofie" is a dog-centric mini-project - it consists of the first script for a Snap! workshop ("STEMonstration") at the Lyon College Academic Day March 4, 2023. Full presentation at <https://bit.ly/3mIENeA>

Created March 4, 2023 Last updated March 4, 2023

Share Publish

Edit Download Embed Add to Collection Delete

Any questions?



Animating a character

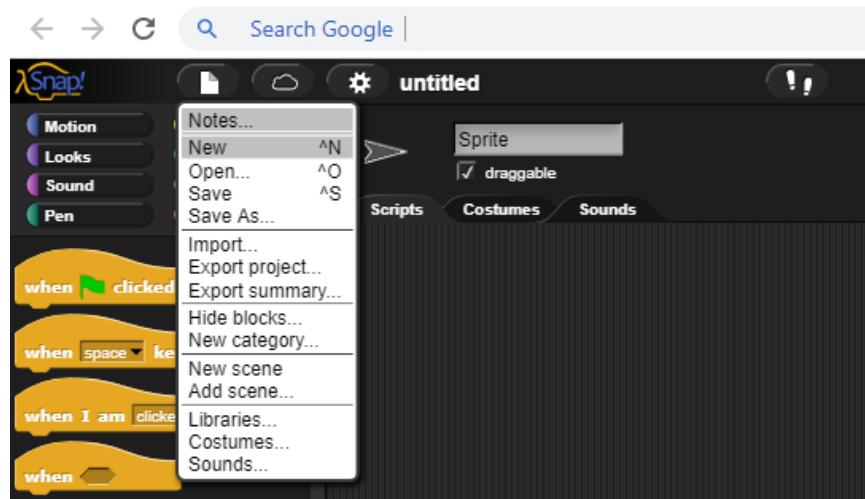


A walk in the park



Final project in the cloud: bit.ly/3J9RAmt

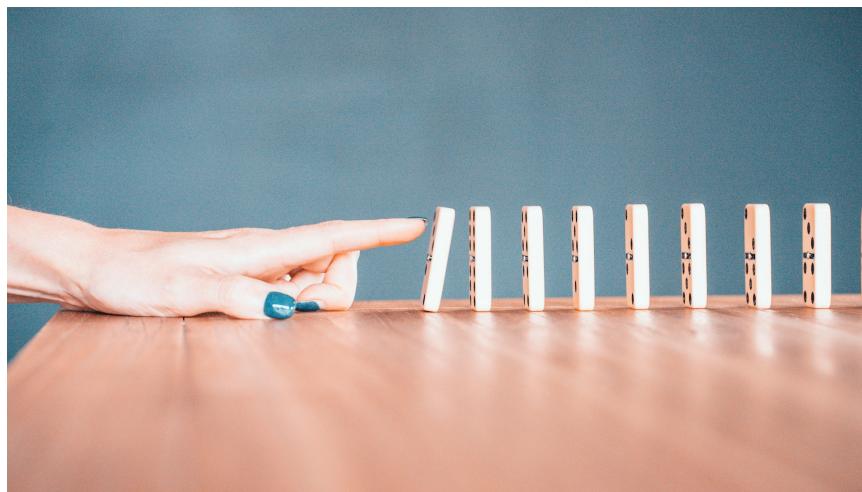
The project



Start a **New** project!

- Open a new project
- Name the project "Avery walking" or "My first animation"
- Add some notes right away (what this is about)

The algorithm



What's the algorithm? Pseudocode:

1. Move object in one direction
2. Swing object's arms and legs
3. If object hits a wall, bounce back

The illusion of walking

`./img/stemdemo_avery_walking.gif`

The sprite



Set object **attributes** (aka sprite costumes).

- Highlight the turtle sprite
- Pick the costumes tab
- Find "Avery walking" sequence
- Highlight one after the other and import them
- Give the Sprite a name "Avery"
- Go to the script tab

The script

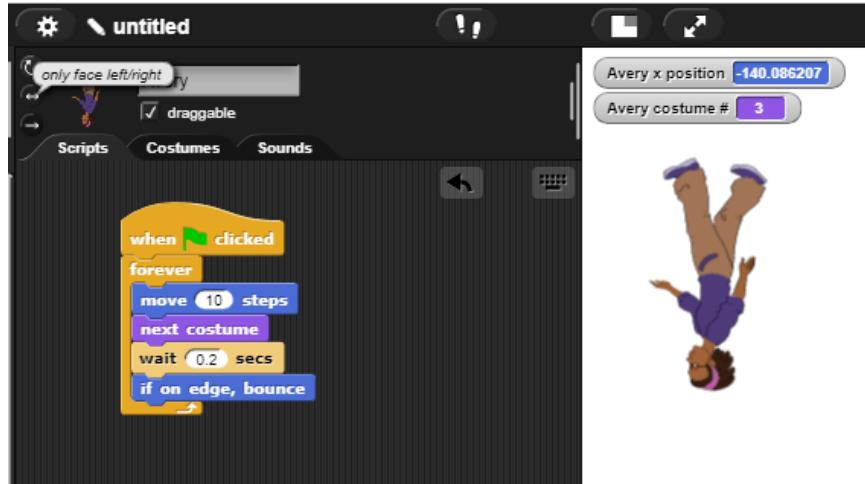


Set object **methods** (aka sprite actions).

- Find 'Control:::forever' and drag it into the script area
- Find 'Motion:::move 10 steps' and drag it into the slot
- Find 'Looks:::next costume' and add it below (with #)
- Find 'Control:::wait 1 secs' and add it below (0.2 secs)
- Find 'Motion:::if on edge, bounce' and add it below
- Find 'Control:::when FLAG clicked' and add it at the top

If you ever "lose" your sprite, right-click on the sprite icon and activate "show", and it will show up again. Better: reset script.

The orientation



Set the sprite to "only face left/right".

The background



- **Download** the park background: bit.ly/3SQaT7F

- **Import** picture as **Stage** background
- Do a final check then **Save** your project
- Final project in the cloud: bit.ly/3J9RAmt

Any questions?



Building a simple game

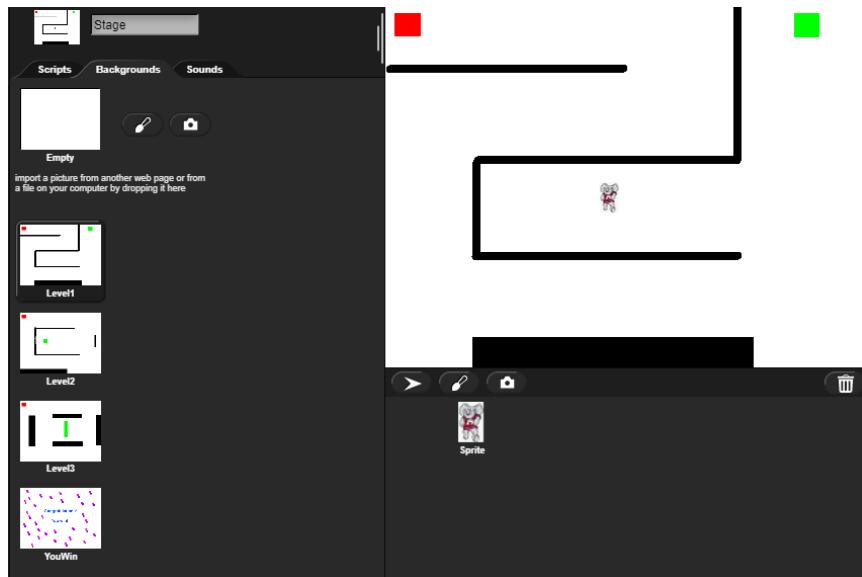


The project

./img/stemdemo_mazegame.gif

Create a new project "Mazegame"

Setup sprite, background, sounds



- 1) Download ZIP file: <https://tinyurl.com/5n8xke8z>
- 2) Extract the file on your PC using File Explorer
- 3) Create new Snap! project and save it as mazegame
- 4) Set up sprite, backgrounds and sounds:
 - Set up (draggable) sprite player
 - Set up sprite sounds for buzzer (wall), clap (level up)
 - Set up stage sound for winning the game (yea)
 - Set up stage backgrounds: three levels and final screen

Make a new Motion block "MovePlayer"



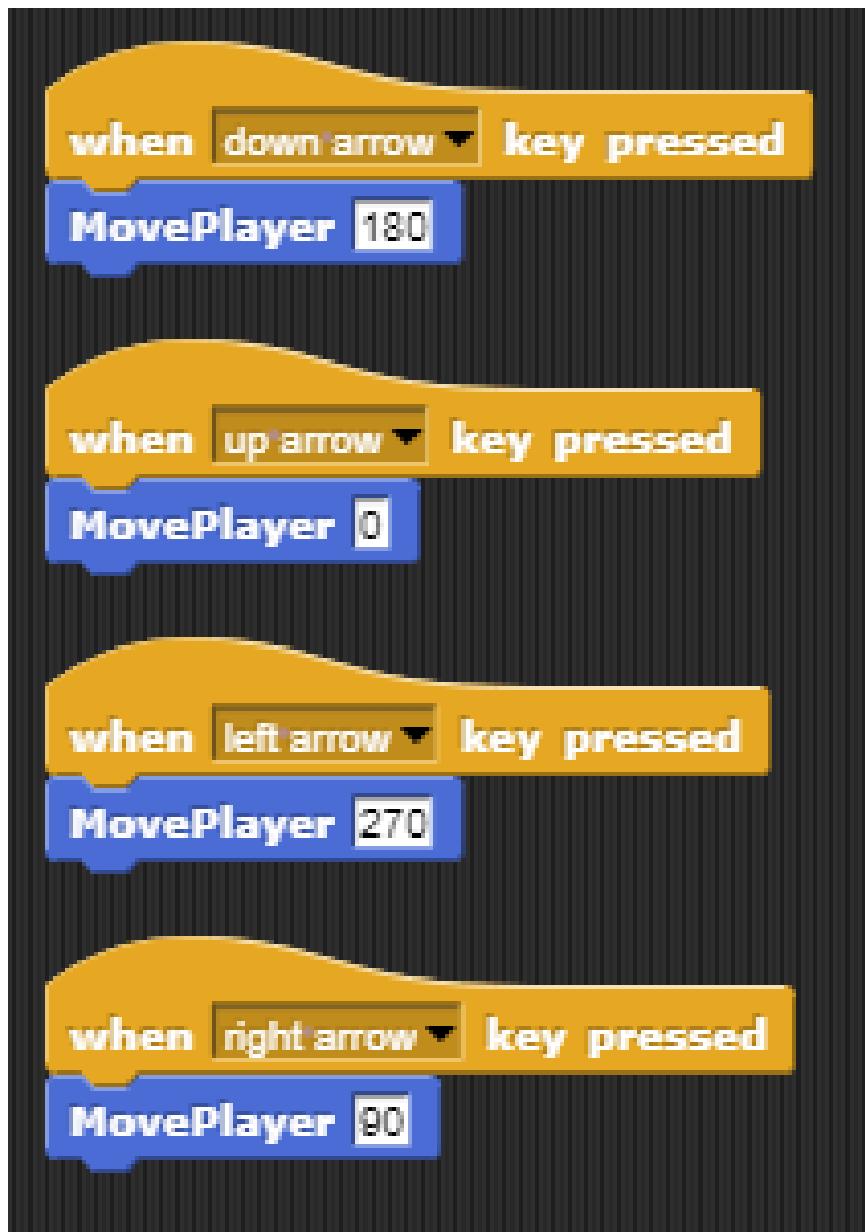
- Open the Motion command block menu
- Click on Make a block
- Enter MovePlayer as Command
- Click on right + sign to add direction
- Apply to all sprites

Make start script



- Make a start script (green flag)
- Point in direction 90 degrees
- Go to starting position ($x = -218, y = 160$)

Set up keyboard action



- Down arrow: 180 degrees
- Up arrow: 0 degrees

- Left arrow: 270 degrees
- Right arrow: 90 degrees

Modify the MovePlayer block



Stage script



When game starts, switch to level1.



- When **next-level** broadcast is received, change costume.
- Play winning sound when all levels are finished.

Refine the game



- Create obstacles for the player
- Add a monster that follows the player
- Add prize points and penalties
- Add a timer and a timeout penalty

Link to the final game

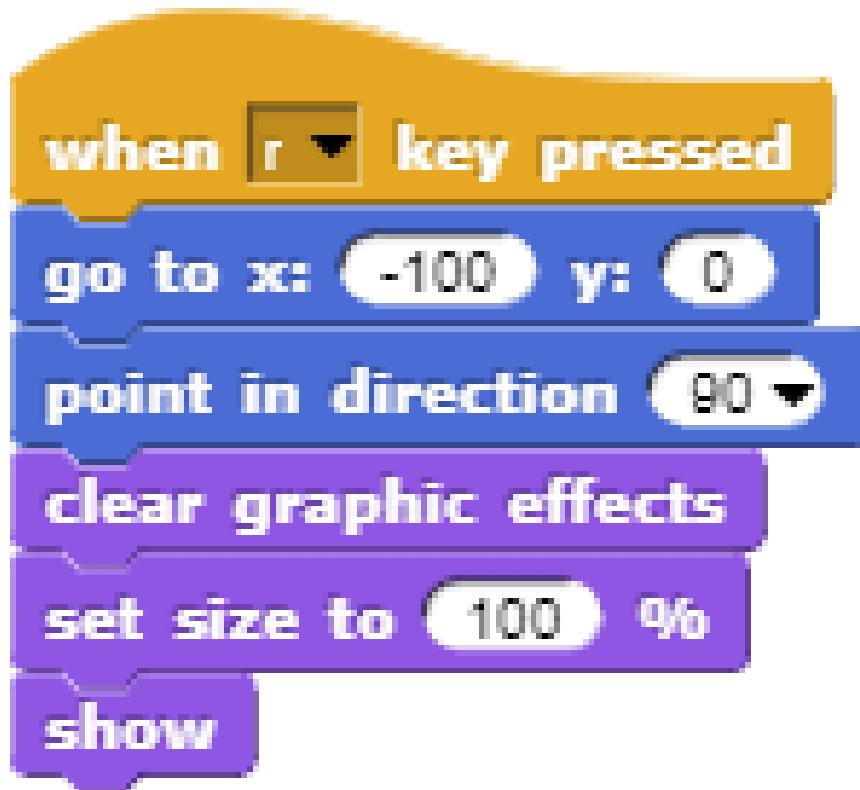
Any questions?



Summary and wrap-Up



Useful reset script



Import this script from

- Start by pressing the 'r' key
- Motion reset: go to center of stage
- Graphics reset: remove effects
- Show sprite

What you learnt (I hope)



- How to use the Snap! UI
- How to write a simple script
- How to make an image move
- How to design a simple game

Further learning



- Birkenkrahe (2023) Lyon College Data Science Program. URL: lyon.edu
- Huegle/Moenig (2020). Get Coding with Snap!. URL: open.sap.com/courses/snap1-1 (Free)

References

- Joshi (2018). Learn CS Concepts with Snap! URL: amzn.to/3IOzsx9
- CSP-Alabama (2014) Mazegame. URL: snap.berkeley.edu