

Coding in Snap! Session

Introduction to Snap!

Presented by Daniel Hickmott

Presentation Contents

- Snap! BYOB
- Website
- The Snap Manual
- Palettes and Panes
- Stage
- Menus
- Turtle Geometry

Snap! BYOB

- BYOB = Build Your Own Blocks
- Modification of *Scratch* with added capabilities
- "These added capabilities make it suitable for a serious introduction to computer science for high school or college students."¹
- Created and maintained by researchers at the *University of California, Berkeley*

¹ <https://snap.berkeley.edu/SnapManual.pdf>

Snap! BYOB

- Like *Scratch*, designed to have:
 - Low floor
 - Wide walls
 - High ceiling
- Free to use
- Does not have the community that *Scratch* does - but still has many examples!

The Website

— snap.berkeley.edu/

Snap! — Build Your Own Blocks

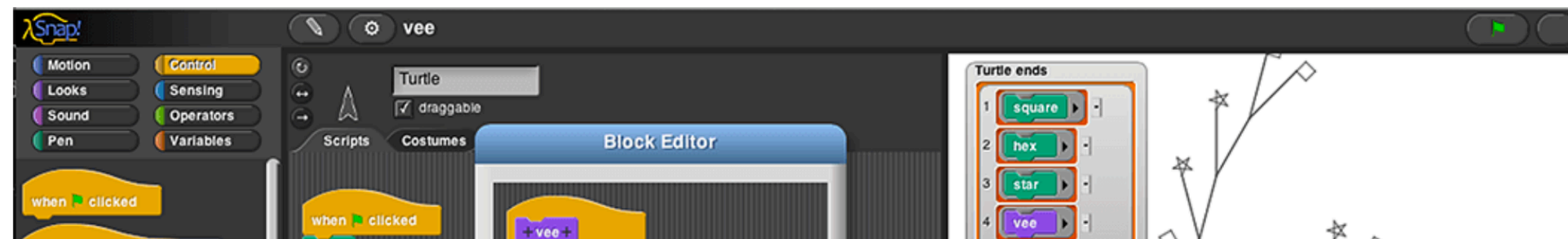
[About](#) [Examples](#) [Teachers](#) [Links](#) [Extensions](#) [Mirrors](#) [Privacy](#) [Thanks](#)



Run Snap! now

What do you get if you combine

- drag-and-drop interface
 - visual metaphors for loops, conditionals, etc.
 - easy animation tools
- from [Scratch](#)
- +
- first class procedures
 - first class lists
 - first class objects
 - first class continuations
- from [Scheme](#)
- ?



The Website

— **Run Snap! Now**

— **About**

— **Examples**

— **Teachers**

The Snap! Manual

- ~90 pages PDF document that explains many of *Snap!*'s features
- Compares features of *Snap!* with *Scratch*

Table of Contents

I. Blocks, Scripts, and Sprites 4

Hat Blocks and Command Blocks 5

A. Sprites and Parallelism 7

Costumes and Sounds 7

Inter-Sprite Communication with Broadcast 8

B. Nesting Sprites: Anchors and Parts 9

C. Reporter Blocks and Expressions 9

D. Predicates and Conditional Evaluation 11

E. Variables 12

Global Variables 13

Script Variables 14

Renaming variables 14

Transient variables 15

F. Debugging 16

B. Writing Higher Order Procedures 42

Recursive Calls to Multiple-Input Blocks 44

C. Formal Parameters 45

D. Procedures as Data 46

E. Special Forms 47

Special Forms in Scratch 48

VII. Object Oriented Programming 49

A. Local State with Script Variables 50

B. Messages and Dispatch Procedures 51

C. Inheritance via Delegation 52

D. An Implementation of Prototyping OOP 52

VIII. The Outside World 56

A. The World Wide Web 56

Snap! Cloud Account

- You can register for a *Cloud Account* to save your *Snap!* projects
- We recommend this approach, unfortunately *Edgy* does not allow this
- Can also save projects to:
 - an *XML file* through *File > Export Project*
 - the *Web Browser* through *File > Save*

Palettes and Panes

- **Block Palette**

- *Scripts* tab

- Blocks are grouped by functionality

- Colours indicate the type of block (for example: all *Motion* blocks are blue)

Palettes and Panes

— **Scripts Area**

- This is where blocks are placed to create *scripts*
- Each *Sprite* has a different *Scripts Area*

— **Sprites Pane**



- Embossed *Sprite* = *Sprite* is selected

Palettes and Panes

— **Costume Pane**

- Also referred to as the *Costume* tab
- Each *Sprite* can have 1 or more *Costumes*

— **Sounds Pane**

- A list of audio clips for the *Sprite*
- Could be sound effects (for example: a dog barking) or background music

Stage

- Also referred to as the *Canvas*
- Has three buttons:
 - *Green Flag*
 - *Pause*
 - *Stop*
- Has its own *Script* area
 - Blocks available to *Stage* differ to those available for *Sprites*

Toolbar

- There is a menu at the top of the *Snap!* interface



- From left to right:
 - *File Menu*
 - *Cloud Menu*
 - *Settings Menu*

Turtle Geometry

— In *Snap!* - *pen down* and *pen up* blocks



Turtle Geometry Project

- You will make a *program* that involves Turtle Geometry
- Finished *Turtle Geometry program* is available on the workshop website
- Familiarise yourself with the *Snap!* environment
- Learn about some fundamental *Coding* concepts
- Involves importing a base project before adding more blocks

Summary

- In this presentation I have:
 - Introduced you to *Snap!*
 - Explained what the main parts of the *Snap!* homepage are
 - Showed you the main parts of the *Snap!* interface
- **Any questions?**