Intro to Snap! Hacking

Michael Ball

- What is Snap!
  - Turtle graphics, inherited from Seymour Papert (~50's / 60's)
  - Demo: vee program
    - File Menu → Open → Examples → vee
- **Snap! Manual -- I forgot to Demo this!**
  - https://snap.berkeley.edu/snap/help/SnapManual.pdf
  - **Extended docs how to do anything in Snap!.**
- Whirlwind tour of features
  - Sprites
  - Blocks, libraries
  - URL block
- Custom Blocks
  - Input types
  - Unevaluated inputs / special forms
- Hacking and Extending
  - JS Function
    - `this` is a sprite
  - Dev Mode
    - I forgot to demo this. In the Snap! Icon in the upper left corner, hold SHIFT and select "enter dev mode" and you'll have some additional tools
    - Morphic Inspector
      - In Dev Mode you can right click and get a custom "Inspector" to view the properties of any object within Snap!.
- Things Needed to Make a New Custom Block
  - Review this PR for a diff: https://github.com/jmoenig/Snap/pull/288/files
  - Objects.js: Register the block in the palette
  - Threads.js: Function that executes the block
  - Blocks.js: Designing a custom menu
  - 2020: Much of this could be done in Snap! Now with a custom JSFunction
- Where to look for hacking:
  - Source locality: Best place is to look at examples.
    - Files are somewhat coupled modules -- they include multiple top-level objects (classes) that are related
  - API.md
  - Morphic.js
    - It's descended from the Smalltalk / Squeak philosophy.
    - The first ~1K lines of the file are an extended code comment/documentation.

Links:

- Homepage: https://snap.berkeley.edu/
- Repository: https://github.com/jmoenig/Snap/
- Manual: https://snap.berkeley.edu/snap/help/SnapManual.pdf
- Curriculum: https://bjc.edc.org/ https://cs10.org
    - https://beautyjoy.github.io/bjc-r/cur/programming/python/introduction_to_besides_blocks.html?topic=berkeley_bjc%2Fpython%2Fbesides-blocks-welcome-parsons.topic&course=cs10_fa20.html&novideo&noreading&noassignment
    - Introduction comparing Snap! To Python
- Cloud Backend (probably not relevant, but fyi): http://github.com/snap-cloud
- Hacking Examples
    - https://github.com/jmoenig/Snap/pull/288/files (Date block)
    - Forks
        - Bad code, but productive ideas: Michael's Comp Photo Hack -- though many things have been built-in now.
    - See the list of extensions: http://snap.berkeley.edu/extensions
- Research and Publications: https://snap.berkeley.edu/research

Miscellaneous and Some Ideas

- Dictionaries / Maps have no clear user representation
    - The Snap-graphics project has a KV representation, extending the list design.
- Lots of possible data types don't have distinct representations, and those could be built.