



# Bringing Real-Time Collaboration to Visual Programming



**Brian Broll**

- Block-based languages have proven effective educational tools
- Collaboration is important in computer science
  - Pair programming
  - Team projects
- However, collaboration is limited
  - Only a single programmer coding at a time
  - Text editing supports multiple simultaneous programmers
- **Goal: Facilitate active collaboration in block-based languages**
- Challenges:
  - What exactly should be synchronized when collaborating?
  - How should undo behave in a collaborative setting?



# What exactly should be synchronized when collaborating?



The screenshot shows the NetsBlox web editor interface. The browser address bar displays <https://editor.netsblox.org>. The project title is "GuessTheMovie @ Movies". The interface includes a left sidebar with various tool categories (Motion, Control, Looks, Sensing, Sound, Operators, Pen, Variables, Services, Custom) and a central workspace with a script editor and a stage.

The script editor shows the following code blocks:

```
when clicked
  set movie_ids to
  list
  158852 11 603 5255 330459 78 13 557 10681 137106 246655 98 155 597 857 12
  1372 333352 129 1781 424 1640 377 1900 87101 348 329865 155 680 274 238
  120 550 807 27205 105 539 492
  go to front
  broadcast start

when I receive check answer
  if movie_id = movie matching answer
    say Yay! You figured it out! for 2 secs
    stop all
  else
    set score to score - 1

when I receive start
  set movie_id to
  item pick random 1 to length of movie_ids of movie_ids
  say Hello! I'm Zara. Can you guess what my favorite movie is? I'll give you hints and you can make guesses (or just pass), one at a time for 5 secs
  set score to 10
  ask join It was released on release date of movie_id and wait
```



# What exactly should be synchronized when collaborating?



- Our approach:
  - Google Docs-style real-time collaboration
  - Synchronizing the source code only
    - As opposed to the execution state (stage)
- Alternative approaches:
  - Synchronize the execution state?
  - Synchronize entire editor state?
  - Simple screen sharing with one person “driving”?



# How should undo behave in a collaborative setting?



- Our approach:
  - Sprite-based (and tab-based) undo
  - Each undo queue is shared between users
    - Basically, the history of the focused content
    - Behaves the same as if built by a single user
    - Users can undo each other's edits
- Alternative approaches:
  - Per user undo queue:
    - May introduce invalid undo actions
    - Undoing in different order may result in a different program
    - Block undo if not the given user's action?