

Bringing Real-Time Collaboration to Visual Programming



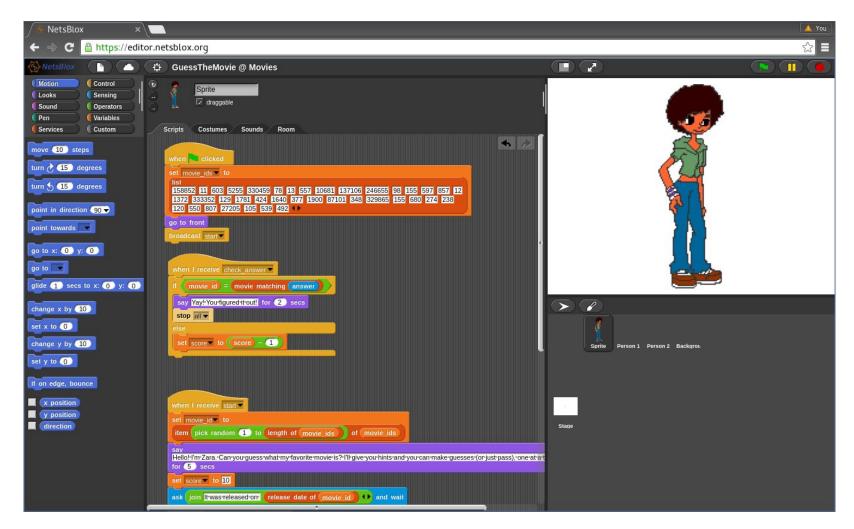
Brian Broll

- 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?







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?