

# Autograding For a Blocks Language

true

true

true

## ABSTRACT

This work describes a novel environment for autograding a visual programming language, Snap!, and its use in a MOOC. We present our design approach, describe some unique features of our system and give notes for how others could implement our work.

Notes - Diffing the “world” State - detecting script changes  
- Ambiguous States or Errors - infinitely looping code - JS Errors - Static analysis - Assertion-Based Testing - following an event log - image diffing

## TODO REFERENCE LIST:

- s.b.e
- bjc.b.e
- Dan Inroads BJC (2012, newer?)
- Colleen’s MOOC
- edX reference
- code.org
- DrScratch
- HairBall paper
- scratch.mit.edu / paper > Mitchel Resnick , John Maloney , Andrés Monroy-Hernández , Natalie Rusk , Evelyn Eastmond , Karen Brennan , Amon Millner , Eric Rosenbaum , Jay Silver , Brian Silverman , Yasmin Kafai, Scratch: programming for all, Communications of the ACM, v.52 n.11, November 2009 [doi>10.1145/1592761.1592779] > Brian Harvey , Daniel D. Garcia , Tiffany Barnes , Nathaniel Titterton , Daniel Armendariz , Luke Segars , Eugene Lemon , Sean Morris , Josh Paley, SNAP! (build your own blocks) (abstract only), Proceeding of the 44th ACM technical symposium on Computer science education, March 06-09, 2013, Denver, Colorado, USA [doi>10.1145/2445196.2445507] > B. Harvey and J. Möning. Bringing “No Ceiling” to Scratch: Can One Language Serve Kids and Computer Scientists? In Constructionism, Paris, France, 2010. > John Maloney

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

, Mitchel Resnick , Natalie Rusk , Brian Silverman , Evelyn Eastmond, The Scratch Programming Language and Environment, ACM Transactions on Computing Education (TOCE), v.10 n.4, p.1-15, November 2010 [doi>10.1145/1868358.1868363]

## FIXME

In this paper, we present the design and use of an in-browser autograder for a visual programming language, Snap! [REF]. Snap!, Build Your Own Block is a web-based, blocks-based language inspired by Scratch [REF], for the course *The Beauty and Joy of Computing*[REF], at UC Berkeley. Before planning to build a MOOC of BJC we needed a way of evaluating student work that didn’t involve human grading. We designed the autograder system that we could have a wide set of unit tests as well as more general assertion tests based on static analysis or output logs. This fall, we successfully deployed our autograder to edX, and piloted it UC Berkeley. We are planning to build out the tools to make them easier for others to use.

## INTRODUCTION

Despite the growing popularity of visual programming languages, there has been little, if any, work published about how to implement any form of automatic grading or assessment for visual environments. We present an autograding tool for Snap![1], which runs in the browser and provides realtime feedback to students. This environment can be used both a feedback tool or as a grader depending upon the configuration and includes functionality for logging students’ test outcomes as well as any code.

## BACKGROUND

## SYSTEM OVERVIEW (?)

## FEATURES

### Data Collection

We aren’t actually testing things, but we are collecting data!

### Adaptability

## IMPLEMENTING THE TOOLS

## FUTURE WORK

## **CONCLUSION**

## **References**