

Autograding For a Blocks Language

AUTOGRADING FOR A BLOCKS LANGUAGE

Meta: Attempting to follow a scholarly markdown format. . .

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

Abstract:

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. ##
Introduction

Lack Related Work (?)

Methods (?)

Writing Test Cases (?)

Data Collected.gitignore

.travis.yml img Makefile paper.md README.md templates
(?) We aren’t actually testing things, but we are collecting data!

Future Work

Conclusion

References

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.