Programming Behavior Detection 1.1

Chao-Yang Lo

COMS E6901, Section 28, Spring 2018

Abstract

Intelligent Tutoring system of Social Addictive Gameful Engineering (SAGE) project has been partially implemented. However, the Scratch Analyzer, the main software to detect and collect user behavior data, has not yet been fully integrated to current SAGE node. Therefore, all detection data is stored and analysis on local machine. The objective of this project is to integrate and deploy the current Scratch Analyzer to SAGE system so that not only the whole Intelligent Tutoring system can be run on server but also it provides a better workflow for the Hinting system. Moreover, this project aims to implement a researcher mode so that the researcher can reach behavior detection system via Affinity space.

1. Introduction

Motivation

The main objective of this project is to enhance the Intelligent Tutoring system of Social Addictive Gameful Engineering (SAGE) [1] project by integrating the behavior detection functionalities. Concrete tasks include the integration of existing Scratch Analyzer, implementation of behavior data persistence, and the implementation of Researcher control system.

Computational Thinking

As of 2006, Prof. Jeannette Wing has addressed the importance of computational thinking [2]. Computational thinking is considered as a fundamental skill which involves solving problems, designing systems. Therefore, computational thinking should be required as one of the every child's analytical abilities.

Scratch

Scratch [3] is a visual programming language developed by MIT Media Lab, designed for 8 to 18 year-old children to develop interactive stories, games and animations. The graphical interface of Scratch provides a set of figures, actions and sound effects. It encourages children to share, reuse and combine different elements to create their own stories and develop ability of computational thinking at the same time. By applying FLASH player on the browser, children can run their program and immediately animate their stories.

SAGE

SAGE is designed and developed to provide a game-like environment for students at grade 6-8 to develop computational thinking skill [1]. The adaptive features and feedback system

can help students optimize their learning experience and hence get “addicted” as other non-educational video game. Also, SAGE also provides a set of functionalities for teachers and therefore can equip teachers with the resources to help students succeed. SAGE is a web application implemented in Node.js. SAGE Assessment Server, or SAGE Node, is the main part of SAGE functionality engine. It provides the assessment of student performance and displays learning related statistics.

Scratch Analyzer

Scratch Analyzer is another major component of SAGE project. It analyzes the log file recorded by SAGE node in a JSON-like format called se file. It is implemented with Java. Also, some Python script is implemented to produce certain learning behaviors such as Mover, Stoppers and Tinkerers.

2. Related Works

Modeling How Students Learn to Program

Piech et al. [4] proposed a series of program distance metric of program snapshots from different students according to natural language processing based features, API call patterns and abstract syntax tree of compiled code. A modeling progress is also proposed to provide a Hidden Markov Model (HMM) assumption when analyzing data. With machine learning algorithm, they were able to recognize the students’ learning patterns with clustering and HMM estimation with EM algorithm.

Using learning analytics to assess students’ behavior

Blikstein [5] analyzed students’ coding behavior by processing programming logs in XML, which is produced by the modified IDE provided for studying to code a programming

task. The log contains information about users' actions including key presses, button click, compilation attempts etc. He proposed a canonical coding strategies by observing action patterns and therefore was able to summarize students' programming profile. Also, he analyzed the relation of compilation attempts and code size is a inverse parabolic shape.

Gameplay as Assessment, Analyzing Event-stream Player Data

Owen et al. [6] proposed a game-based assessment model, which might be used as any game-based learning research project. This includes two parts: Game design process and Assessment process. Game design process defines the main component and game-flow of their game about regenerative medicine. Assessment process defines the indicators of real learning and the analysis of players' profile. Also, a pre-post assessment test is the main evaluation tool for quantifying the gain of knowledge learned.

3. Proposal

Scratch Analyzer to SAGE node connectivity

Currently, both Scratch Analyzer and the Python script to generate behavior profiles are run on local machine. I will first evaluate the feasibility to run both on SAGE node and then implemented the communication between SAGE node and Scratch Analyzer.

Programming Behavior Persistence

The persistence of behavior detected data is at best implemented on server. By doing so, student will be able to experience all the features affected by Intelligent Tutoring system once both data can be collected and analysis can be conducted on the server.

SAGE-Scratch Integration

By further integrating SAGE and Scratch Analyzer, students' behavior detection and analysis can be fully automatic. Therefore, teacher will be able to identify students' programming behavior and students will be assigned a certain category of profiles.

Behavior Modeling as via Researcher Role

The implementation of back side of behavior detection system will allow the researcher to manipulate the behavior model via Affinity Space interface instead of manually modifying scripts. This will largely enhance its application in research since researchers without programming experience will have the chance to adjust the models.

4. Timeline

Milestone	Estimated Dates
Project Proposal	February 2, 2018
SAGE Scratch Setup	February 9, 2018
Scratch Analyzer Deployment	February 23, 2018
Behavior Data Persistence	March 9, 2018
Midterm Report/Presentation	March 23, 2018
SAGE-Scratch Integration	April 6, 2018
Researcher mode Implementation	April 20, 2018
Final Report/Presentation	May 7, 2018

References

1. J. Bender, *"Tooling Scratch: Designing a Collaborative Game-Based Learning System to Infuse Computational Thinking within Grade 6-8 Curricula"*, Columbia University, 2015
2. J. M. Wing, "*Computational Thinking"*, Communications of the ACM, 2015
3. Scratch. Retrieved from https://scratch.mit.edu/
4. Piech, Chris, et al. *"Modeling how students learn to program."* Proceedings of the 43rd ACM technical symposium on Computer Science Education. ACM, 2012.
5. Blikstein, Paulo. *"Using learning analytics to assess students' behavior in open-ended programming tasks."* Proceedings of the 1st international conference on learning analytics and knowledge. ACM, 2011.
6. V. E. Owen et al. "*Gameplay as Assessment: Analyzing Event-Stream Player Data and Learning Using GBA (a Game-Based Assessment Model).*" International Conference on Computer Supported Collaborative Learning, 2013