

# To Introduce Computer Science in One Day, The Throw Platform

Joseph Paul Cohen, Wei Ding, and Deborah Boisvert

University of Massachusetts Boston

Boston, MA 02125–3393

Email: {joecohen,ding}@cs.umb.edu, deborah.boisvert@umb.edu

**Abstract**—This paper presents an open source platform Throw [1] that is intended to be used for a one day event to increase interest in computer science as well as equip students with tools for self exploration. It is based on three years of hosting a one day event aimed at introducing computer science to 5-8th grade students. One major challenge in computer science education is fostering a commitment to learn, that is powerful and engaging enough for a student to be able to utilize learned skills to turn their ideas into reality and drive them to pursue computing. We present a theory for what is necessary for an effective introduction to computer science as well as present survey results from the first trial of the Throw platform which encompasses this theory.

**Keywords**—STEM Education, Computer Science Education, Web

## I. INTRODUCTION

One major challenge in computer science education is fostering a commitment to learn, that is powerful and engaging enough for a student to be able to utilize learned skills to turn their ideas into reality and drive them to pursue computing. We have faced this challenge for the last three years while introducing computer science to 5-8th grade students as part of a summer camp that is a collaborative effort with Boston area universities.

This paper presents an open source platform Throw [1] that is intended to be used for a one day event to increase interest in computer science as well as equip students with tools for self-exploration. Based off previous years events we believe that the following criteria are the key to an effective introduction to computer science:

- *Instant Graphical Feedback (IGF)*: Small amounts of coding result in dramatic changes to the resulting creation. To encourage trial and error programming and experimentation.
- *Collaboration Ability (CA)* The ease of both allowing others to experience a creation and allowing others to make derivative works. To reduce the the barrier to dissemination and the complication of modification.
- *Industry Applicability (IA)* How related are these skills to what is currently used in industry. Can this be used to build real applications.

It is difficult to recruit students for a semester long course if they are not already interested in computer science. A student may be more willing to attend a one day event in order to

	Programming Style	Language (Most Similar)	Runtime Platform
Throw [1]	Textual	HTML/Javascript	Browser
Alice [2]	Visual	Karel	Python
Processing [3]	Textual	Java	Java
Scratch [4]	Visual	Alice	Java

TABLE I: Comparison of CS0 Teaching Platforms

gauge their interest. However, a one day long event requires a teaching tool with a very short learning and development cycle. For this, Throw is a visually powerful tool to engage and create aspiring computer scientists.

We believe that IGF is an important aspect of software or this purpose. Cooper in 2000 suggested that visual feedback was important in the teaching tool Alice [2] when stating that “Alice can be used to follow Karel’s tradition with a 3-D, animated environment where students can create their own virtual worlds.”

In past events, Processing [3] was very successful because students were able to make small alterations to code and achieve instant feedback. Processing includes an ArcTangent example that generates a window with eyes that follow the mouse cursor. By modifying the source code slightly the eyes can change size, move around the window, or even look away from the cursor.

With Throw we also aim to provide CA by making it easy to share and make derivative work such as how Scratch [4] provides a website to exchange creations. There is a notion that being a computer scientist is not a social activity. Students may want to avoid this field because they believe being a computer scientist would be secluding themselves. Using GitHub provides the social coding engine to achieve this as it is actively used by most open source programmers to collaborate on cutting edge projects.

IA is an important aspect because if students only learn a few things about computer science they should widely applicable. The foundation of Throw is HTML/Javascript which is the foundation of the Internet. Learning these basics will change the way the students look at the world they see everyday.

It is the hope that further reaching implications of the Throw platform will yield more students into the STEM fields as well as give students a greater sense of independent innovative thinking coupled with the abilities required to execute ideas independently.

## II. RELATED WORK

There are many excellent existing applications shown in Table I on which Throw was based on.

Alice [2] [5] presents users with a self contained GUI where they drag and drop elements of a visual language to create a program that controls a 3D scene of various objects. This platform achieves the goal of IGF because small changes via drop downs and inserting for loops easily affect the 3D scene. Alice however does not provide CA because it is a desktop application that must be installed making it difficult to showcase work for people without Alice installed on the computer/device. In terms of IA it provides a basis for 3D programming paradigms but learning the Alice platform is a task in itself that does not translate to real world applications.

Scratch [4] is a tool very similar to Alice in the way the user edits the creation in both development style and language. In terms of CA, scratch makes significant advancement by making it easy to share creations from a desktop computer to the scratch website. The published creation requires the web browser to launch an Adobe Flash plugin which is unavailable on platforms such as mobile phones, tablets, and Linux.

Processing [3] was initially designed as a platform for biologists to create visualizations from data. A textual programming language, derived from Java, is used in a self-contained editor with a simple “run” and “stop” interface. Examples provided with the platform allow excellent jumping off points for altering existing code. This platform provides more IA because it is based off Java. Processing provides IGF via modifying example code to change values such as size and color. The CA of Processing is good but it requires users to download and install software. Sharing creations is possible because HTML files can be exported, but hosting on a website is nontrivial and it requires a Java Applet plugin in a browser which is unavailable on platforms such as mobile phones and tablets.

## III. THE THROW PLATFORM

Using Throw [1], with a small amount of textual programming, students are able to create an interactive touch screen game that is culturally relevant to them and their peers. This game is edited in a web browser and is then accessible via a specific URL. This URL can be shared with others who can then view the source of that page to incorporate it into their pages.

There are two major parts of this platform, hosting/editing and the Throw template. The template is simply a basic HTML/Javascript page with the inclusion of a custom version of the open source GravityScript [7] which allows for explicit labeling of objects as the following types of objects:

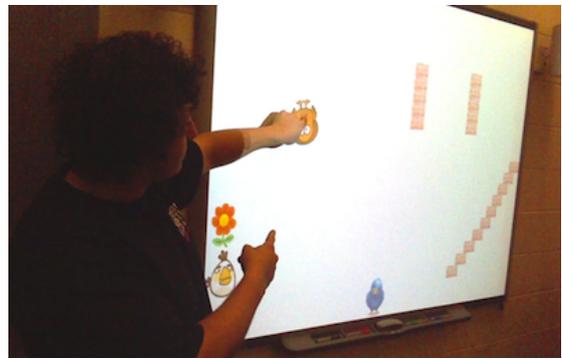
- **grav** - Object is affected by gravity, impacts other objects, and can be moved with the mouse or touchscreen.
- **solid** - Object is not affected by gravity and does impact other objects but are stationary. This would be used on the walls of maze.
- **No label** - Objects are visible but don't affect *grav* or *solid* objects.

The main task of the student is to layout images using *top*, *left*, *right*, *bottom* locations specified by CSS style tags. For example the following is the basic element in Throw:

```

```

By using a standard HTML image tag the student can use knowledge they learn from many available sources to extend their Throw creation. Each element becomes it's own solid object that maintains rotation and velocity. For example in Figure 1a the turkey is a *grav* element and is being thrown towards the basket which is a collection of *solid* objects. In Figure 1b each letter of the word THROW is labeled with the *grav* class and responds to being dragged around. On the tablet the gravity that normally pulls objects down reacts to changes in device orientation and causes elements fall to whatever “down” is on the device. With these basic constructs students can create unique personalized creations.



(a) A demo of the Throw platform on a touch screen presentation board. In this game you throw objects into a basket made of bricks.



(b) Throw on a tablet web browser using the touch screen as input. Throw can read the orientation sensors and adjusts gravity to pull towards what way is “down” on the device.

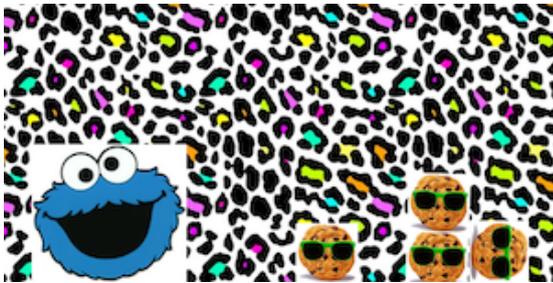
For the first deployment of this platform GitHub was used as hosting and editing. Students are provided a website at [username.github.com](http://username.github.com) that they can edit using a web based editor. Students can elect to follow their peers activity and are alerted by the site when edits are made. This creates a social environment for the students involved. Another positive about this solution is that GitHub is not required so instructors may use a hosting/editing solution of their choice.

These creations can be shared via a link, sent in an email as an html file, or shared via social websites. These files are easy to modify and derive new work on any computer. Only a web browser is required.

In terms of IGF, changing an image by linking to another one or relocating an image on the page will drastically change the meaning of the creation. For CA, Throw's programs are simply text files that are run or edited. These files are static web pages and require no server side processing. The current hosting/editing solution allows for each student to share their creation and have the recipient view and modify the creation with only a web browser.

Throw is based off industry platforms such as HTML and Javascript. The method in which the student works with the code is very similar to that of an industry web designer which gives it high IA. The use of GitHub as the hosting and editing platform is also highly related to industry collaboration tools.

In Figure 1, creations from the 2012 first trial of Throw are shown. These were created during the course of two hours with basic instruction on how Throw works, how to get images from the internet, and basic CSS syntax.



(c) The Cookie Monster attacks a pile of cookies



(d) Mickey Mouse and Minnie Mouse are thrown around with hearts that bounce off each other

Fig. 1: Throw creations that were created during the first trial of Throw

#### IV. CONCLUSION

Surveys were used during the first trial of Throw. The results are very good. In Figures 2 and 3 higher numbers indicate a more positive response. Figure 2 demonstrates that almost all students felt they had a greater understanding of computer science. Not only do they feel confident that they know what it is, they also are now very interested in computer science as seen in Figure 3.

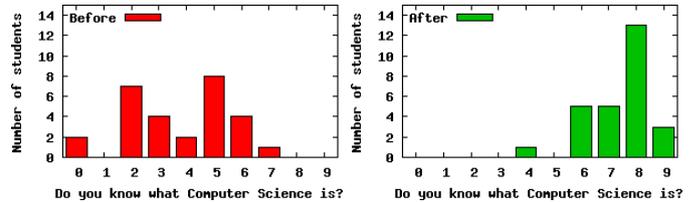


Fig. 2: Before and after survey results for the question "Do you know what computer science is?"

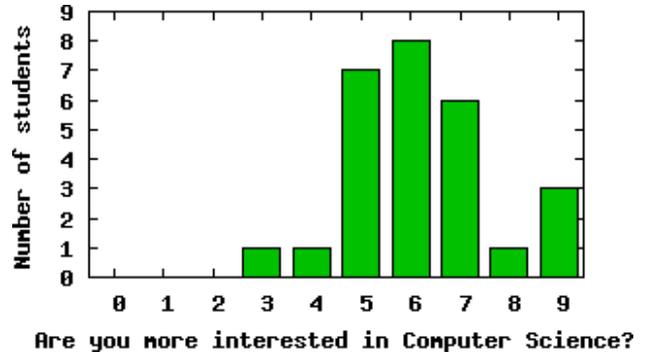


Fig. 3: Survey results for the question "Are you more interested in Computer Science?"

#### V. FUTURE WORK

There is much more research and modification needed to make the Throw platform the premier tool for Computer Science education. A dedicated hosting/editing solution would allow the interface to be more intuitive because it wouldn't have to rely on the GitHub website. An ideal solution would have an online editing interface that assists the student as well as provides a side by side view with source on one side and resulting creation on the other. The process of publishing could also be made easier by having a simple share button to generate a URL that can be sent to other students.

#### REFERENCES

- [1] J. P. Cohen, "Throw! (the game platform)," <http://throw-platform.github.com/>, 2012. [Online]. Available: <http://throw-platform.github.com/>
- [2] S. Cooper, W. Dann, and R. Pausch, "Alice: a 3-d tool for introductory programming concepts," in *Journal of Computing Sciences in Colleges*, vol. 15, 2000, p. 107116. [Online]. Available: <http://dl.acm.org/citation.cfm?id=364161>
- [3] C. Reas and B. Fry, "Processing: programming for the media arts," *AI & Society*, vol. 20, no. 4, p. 526538, 2006. [Online]. Available: <http://www.springerlink.com/index/d0g5745510252632.pdf>
- [4] M. Resnick, J. Maloney, A. Monroy-Hernandez, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: programming for all," *Commun. ACM*, vol. 52, no. 11, p. 6067, Nov. 2009. [Online]. Available: <http://doi.acm.org/10.1145/1592761.1592779>
- [5] C. Kelleher, R. Pausch, and S. Kiesler, "Storytelling alice motivates middle school girls to learn computer programming," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2007, p. 14551464. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1240624.1240844>
- [6] K. Powers, S. Ecott, and L. M. Hirshfield, "Through the looking glass: teaching CS0 with alice," *ACM SIGCSE Bulletin*, vol. 39, no. 1, p. 213217, 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1227386>
- [7] J. P. Cohen, "gravityscript - javascript to make objects on your webpage obey gravity, by joseph paul cohen," <http://code.google.com/p/gravityscript/>, 2011. [Online]. Available: <http://code.google.com/p/gravityscript/>