**CS 490**

*Open Click*

*Ben Waters, Ryan Wolfe*

*4/2/2015*

**ABSTRACT**

The aim of Open Click is to provide teachers and students with an easily accessible and usable way of live classroom quizzing at an extremely low cost especially for the latter. Open Click is completely open source and customizable to the teacher, student, or institution. Currently, we are unable to report on any findings or conclusions since we are still in the process of completing *Open Click*. Later down the road, when we are finished, we will be able to summarize our findings and conclusions in this paper.

**INTRODUCTION**

Currently, there exist multiple implementations of live quiz systems such as iClicker and Top Hat Monocle. The one thing that both of these systems lack is expendability and a low cost to students and teachers. iClicker and Top Hat Monocle both require the student to pay for the usage of the system. iClicker requires that the students purchase individual hardware to access the system and Top Hat Monocle requires that the students pay a fee to sign up and access the website that runs the software. Our implementation, *Open Click*, will be free for the students. We set up an easy to configure environment built into the cloud with an option of open source hardware using a Raspberry Pi. *Open Click* is essentially a CMS (Content Management System) designed specifically for education and learning.

**RESEARCH QUESTION**

Can you make a free and simple, yet customizable and useful live questioning system for classrooms to replace current implementations such as iClicker and Top Hat Monocle?

Currently, of the most common implementations of live questioning systems for classrooms, such as Top Hat Monocle and iClicker, neither of them are particularly desirable for the student or professor. Both have their pros and cons, but with *Open Click*, we aim to eliminate as many cons as possible and make an implementation that is sought-after to everyone.

Neither of the two implementations are free. Both systems require the user, the student, to pay for either a device or subscription. We are aiming to eliminate the need for money by allowing the user to be able to answer the questions provided by logging into the website, through Mason, by any device that allows them to connect to the internet. *Open Click* will also allow for some degree of customization specific to each user. *Open Click* will allow for the professor to create questions that differ in variety. The professor can create questions such as multiple choice, short answer, fill in the blanks, essay, tables, charts, an a multitude of others. This differs from Top Hat Monocle and iClicker because both of those implementations only allow for multiple-choice types of questions.

**PROBLEM STATEMENT**

One of the many problems that we must tackle with this project is that *Open Click* must be easily accessible with low Internet accessibility. The way that we are going to handle this issue is by utilizing a Raspberry Pi as a base station. If the classroom or location has poor Internet, the information will be sent to the Raspberry Pi and stored locally. Once an Internet connection is available, that information will be uploaded to the cloud.

*Open Click* must also have a low cost for its use. As mentioned above, many implementations of this concept require a fee from the student for the usage. *Open Click* strives to make the use of the system cost free for students. If a university wants to utilize the *Open Click* software, a fee may be required from the university just for maintenance of the website. Also, if the teacher or university knows that they are going to be utilizing the system in an environment where there is no Internet, they must purchase a Raspberry Pi to use as the base station. The free use for the students must be one of the primary goals.

Besides a low cost and easy accessibility, *Open Click* needs to be extendable and customizable, both in terms of interfaces and questions. Both iClicker and Top Hat Monocle only allow for the use of multiple choice type questions. For *Open Click*, our goal is to allow the professor or main user of the system to choose the question types from a more expansive list of options. The problem that we will run into is how to allow multiple types of questions and how we will create the structures for them.

Besides the 3 primary focuses that we must tackle, we must also try to make sure that *Open Click* is simple in that it is easy for the students to use, is mobile friendly, and is easy to import and export data. We also aim to implement an attendance-taking feature to *Open Click*.

We aim to also implement an aesthetically pleasing website design. Having a website layout that is aesthetically pleasing to the eye and easy to navigate goes hand in hand with allowing us to reach our goal of *Open Click* being a simple, user friendly implementation. Currently, mock-ups of website layouts and designs are being worked on using Adobe Photoshop and Adobe Illustrator. When the layouts and designs are complete and we are satisfied with how they look and we feel that they will contribute to our goal and research question, we will then proceed to create those layouts in code using CSS and implement them into *Open Click.*

**SOLUTIONS**

In terms of information gathering, we did research into iClicker and Top Hat Monocle and the features and implementations of them. We compared and contrasted the features of both of those systems with the goals and ideas that we have for *Open Click* and looked at what we should focus on to make *Open Click* stand out and be better than the others.

To implement the cloud, our goal is to set up Ubuntu 14.04 and run a Nginix and Django Webserver. A hardened security system will be used through 443/SSL that only allows keys and approved users.

For implementing the Raspberry Pi, we will use a Raspberry Pi b+ running Django Webserver and have it connect to an LCD display for visualization. The Raspberry Pi will have Bluetooth set up in order to receive incoming information. The teacher will create questions on the local Raspberry Pi and will sync to the cloud once an Internet connection is received.

For the next time that we meet in class, for practice demos and presentations, we are aiming to have a reliably working demo with most features working. The website should be up and running with all necessary files and resources so the demo can be run there instead of locally. Others in the class should be able to login and be asked questions.

**CONCLUSION**

We are not able to conclude on any results or findings at the time since we are still working on finishing our implementation. At the end, when we finish, our results as well as discussion as to if we answered our research question or not will be posted here.

**REFERENCES**

**APPENDICES**