**Fall**

16

**725 23rd Street NW Washington DC 20052**

Project Jabberwocky

Lucas Chaufournier

08

**Fall**

# Overview

Currently students new to programming have two options for environments they can use to program: text editors and Integrated Development Environments (IDE). Both of these options often hamper a new students ability to learn programming. Text editors are too bare bones and offer no assistance to the student as they work their way through beginning material. IDEs provide too much support, overwhelming the student with too many features, and offering too much assistance in debugging, a key skill for new programmers to develop. In IDE’s, features such as auto generation of classes and syntax completion, prevent students from learning the in and outs of a programming language making them weaker developers when it comes to harder content.

Jabberwocky offers a new environment for beginning students to learn software development. Whereas current development environments are very static for new students, the Jabberwocky editor is a dynamic editor that evolves as the students’ skills develop. As students demonstrate they understand a concept such as creating a class, the editor then will unlock the auto generation feature. As a student learns how to debug a program the manual way with print statements, a debugger is unlocked. This will help students learn to problem solve and learn a language without an editor that coddles them or leaves them stranded. A more welcoming environment is provided to the student, fostering their ability to learn.

In a classroom environment Jabberwocky interfaces with a central teachers console, allowing instructors to set limits on features, as well as thresholds for when a skill considered mastered. The instructors console will also allow the collecting of all student data, allowing the editors to learn from each other and make better decisions based on the data. Jabberwocky will identify students that are struggling and students that are doing well and treat them separately. Students that struggle can be provided additional resources while students that succeed can unlock advanced features faster. This will ensure that students are not abandoned as they learn and provides an environment for them to feel comfortable learning

Jabberwocky focuses on providing individual support to students, only unlocking features as individual students master a concept, functioning more in assistance as opposed to a crutch. Due to the individualized nature of how students learn, it will require a system to observe user habits and make decisions based on this. The backend of Jabberwocky will rely on machine learning techniques to learn about students and will individually determine whether they are ready for a new feature.

# Business Plan

Jabberwocky has the unique opportunity to take over the educational market for programming. There are very few education IDE’s on the market that exist today especially ones that focus on a classroom environment. Most of the current IDEs such as Scratch or Alice are built for the elementary or middle school level and rather than focus on syntax of languages and tools for development they focus just on concepts. This usually involves on drag and drop interfaces that are not constructive for new programmers especially at the high school or college level. These editors also don’t allow an instructor to see the performance of their students and assist the students on their assignments.

Similar to the Blackboard taking over the online classroom management market, Jabberwocky can be introduced to high schools and colleges to manage their introduction programming courses. A similar revenue strategy will be used to turn a profit from Jabberwocky. A low yearly license fee will be charged to the institution for hosting the platform on their own servers with an optional support fee. If an institution such as a high school does not have the resources or infrastructure to host the platform a higher yearly subscription fee with included support can be charged to the institution. We also plan to focus group test a downloaded content model where the product is offered free to the institution. Students will only be allowed to compile five times per hour and the amount of compilations will recharge every 20 minutes or they will have the option to buy additional compilations for $0.99 and 3 for $3.99. They will also be able to skip levels by paying $5.99 per level with the cost increasing per level. Students will also unlock bonuses by connecting to their favorite social networks. We aren’t sure how well this will test with users but it is a revenue option.

Jabberwocky is based off of the cloud9 product that provides a full-featured editor entirely run from the browser from any location. In a recent survey of 1200 cloud9 users, 64% said they would not switch back to a local development environment given the chance. The survey also found that the four main benefits of the cloud9 editor are “Accessing the same environment anywhere”, “Environment Management”, “Collaboration”, “Easy prototyping or learning”. All four of these are important qualities desired in an educational environment and because cloud9 is the basis for Jabberwocky, it makes Jabberwocky a good candidate for success in the education space.

Jabberwocky will penetrate the market space by offering professors a course long demo that includes the full-featured editor. For future semesters, Universities will have to pay the standard contract. This will allow professors time to see whether Jabberwocky will fit their classroom and teaching style rather than having to make a snap judgment. Only one trial license will be allowed per university, thus only one class can experiment rather than multiple trials happening at once. This will allow us to expand in the university space at pace of about 1 semester per University.

Jabberwocky also plans to expand into the Massively Open Online Course (MOOC) market. Currently MOOC’s require students to perform all of their work on their own machines, which can lead to many problems for the instructor since courses are spread out across the world. With Jabberwocky students should be able to perform all of their work in preconfigured environments in their web browser. Jabberwocky will also allow the MOOC instructor to easily view students’ work and with future editions of Jabberwocky, grade their assignments. Jabberwocky will make the distributed nature of MOOCs not feel so distributed. It will enable instructors to take advantage of all the benefits of the distributed classroom without its problems.

While Jabberwocky has the capability to overtake an untapped market space, it will still need to overcome several problems before becoming successful. We need to make sure that we have the infrastructure to support classrooms with student sizes from five to hundreds of thousands of students. If we are to host the infrastructure we will need data center space dedicated to host the platform. We will also have trouble initially breaking into universities that teach languages other than Java. Since Jabberwocky is built around Java we will have to quickly expand to other more popular languages such as C and Python to be able to maximize our market potential. Since Jabberwocky is backed by a small development team we also need to expand the size of the development team and a marketing team if we want to rapidly expand Jabberwocky.

While still small the team behind Jabberwocky is very qualified to succeed in the market space. The lead developer has experience in teaching undergraduates programming as well as working through many development environments. He also is about to enter graduate school where he will be expected to teach undergraduates programming. This experience enables Jabberwocky to function at the needs of both the students and the teachers that instruct them. We approach the product from the view of student and what enables a student to best succeed when learning to program from the first time. This allows us to give the student the best experience possible and to maximize their learning potential. We then approach the product from the view of a professor who has to manage all of their students and make sure everyone succeeds. This allows us to give the instructor the best tools to make sure their students are learning.

# Social Impact

Jabberwocky has the potential to be the de facto tool used in educational programming environments much similar to the way that Eclipse is used as the primary integrated development environment for Java programmers in the workplace. The tools provided by Jabberwocky can help to enable new students to quickly learn the tools they need to succeed in the workplace before they even step through the door. Previously students would learn to program using bare bones editors or way too advanced editors and would not have the time to learn the more helpful tools that are needed for success in the workplace.

Jabberwocky will affect those who have a desire to learn programming the right way. This includes students in a programming course, a MOOC, or an individual who takes advantage of the community edition for individual users. These students will be saved from the pains of a normal programming experience where they are left completely left alone to figure out the material. This has the chance to keep programmers who would otherwise abandon programming, once it got too difficult for them, motivated to continue on. Departments who normally had difficulty retaining students due to the challenging nature of programming, would instead see their students prosper and reach more advanced material.

On a broader impact level, with proper instruction and the capabilities of Jabberwocky, intro programming can be introduced into the standard high school curriculum. The ease of deploying a Jabberwocky setup combined with the affordability, but drastically lowers the barrier to entry of high schools getting involved. With a widespread programming curriculum due to Jabberwocky, society as a whole will begin to be a more technologically savvy population.

For our core infrastructure we need to ensure that our datacenter is environmentally responsible and takes care to be energy efficient. As computations of students code and evaluations of their learning ability may involve a lot of processing power, it is our responsibility to choose a provider who considers the environment. We also have to make sure that we source data centers from around the world so as to follow international data laws on how data can be handled. To save on energy costs, instances that have not been logged into in more than a week will be powered down. The user will be able to start it up again when they log in again.

Jabberwocky takes security very seriously. We insure that code executed by students is restricted to their separate docker instances and can not be executed on the underlying system. We also store our student databases on separate servers so if the underlying system is compromised no data contamination can occur. As a further measure Jabberwocky only collects the bare minimum on student data, just enough to uniquely identify students to their learning profile. System monitors will also be put in place to monitor systems to make sure processes out of the ordinary are caught and suspended before damage occurs.

Jabberwocky has the potential to make a bigger impact on the world, but it can only come with support from the community and investors who believe in our mission to change the world of educational programming.