

Collaborative Research: Type I:

FRABJOUS CS — Framing a Rigorous Approach to Beauty and Joy for Outreach to Underrepresented Students in Computing at Scale

We propose to reach out to students from underrepresented groups by disseminating, researching, and improving our **Beauty and Joy of Computing** (BJC) curriculum. We build on our work as two of the five initial pilots for the Advanced Placement *Computer Science: Principles* course to support the CS10K initiative through in-service teacher professional development and course adoption. We use the Snap programming language, an extension of Scratch formerly known as “Build Your Own Blocks,” that combines technical sophistication with an attractive drag-and-drop interface. ***BJC invokes passion, beauty, joy, and awe through engaging students in a rigorous computing curriculum that promotes creativity and collaboration using Snap’s visually rich programming environment, while also provoking thought around current events and how computing relates to people’s lives.***

Our **primary objectives** are to support the CS 10K and AP CS Principles efforts to prepare both high school teachers and students to be creators in computing through this project to: (1) conduct, evaluate and scale team-based professional development for **100 in-service teachers** supported by **regional collaboration** with university faculty and students, (2) empirically investigate the effectiveness of our curriculum, with particular emphasis on adoptability in high schools and understanding what works best for **underrepresented minorities**, and (3) **enhance the Snap software** with debugging support, and the ability to run in the browser. We will collaborate with the STARS Alliance to inform our efforts to broaden participation and the Computer Science Teachers Association (CSTA) to provide a vibrant **community of support** for high school teachers and students engaging the new BJC course. Activities will include:

Outreach and Professional Development

- Develop a core group of **mentor teachers** starting in Berkeley and Charlotte, who will help scale professional development (PD) to new locations.
- Conduct and evaluate intensive summer professional development workshops and school-year support activities for **100 in-service high school teachers** who will teach BJC in their schools. Workshops will emphasize the BJC curriculum content, including both Snap programming, and the social, cultural, and historic elements of the AP CS principles curriculum, while also promoting lab-centric classroom techniques to promote collaboration, creativity, and sense of identity.
- Develop **regional partnerships** between universities and high schools to implement the BJC curriculum, replicating our practices at Berkeley and UNC Charlotte of building local CSTA chapters and connecting them through the STARS Alliance.

Assessing Curricular Effectiveness

- Study university and high school **student outcomes** (including grades, attitudes, and success in subsequent courses), disaggregating results by race, gender, age, course, and curricular modules to understand the effectiveness of the BJC curriculum, particularly for motivation and learning in students from underrepresented groups. Although we have (mostly anecdotal) evidence of the course’s success at UNC Charlotte and Berkeley, research is needed to verify our findings, and to understand how well each model translates to other environments (university, high school, etc). This will help inform the AP CS Principles and CS10K projects.
- Compare outcomes for our own (university) students, high school students taught by teachers trained directly by the PIs, and high school students taught by second-cohort teachers prepared by mentor teachers (to ensure the **scalability** of our PD program).

Tools Development

- **Expand the capability of Snap** programming language and development environment in collaboration with its lead developer, Jens Mönig, to improve its speed and debugging support, and allow development in a browser without the need for local software installation. Snap is central to the BJC curriculum because of its support for technical rigor in a format that attracts non-CS students.

BJC's curriculum design builds on Berkeley's experience offering Scratch-based summer programs to underrepresented middle and high school students, as well as the experience of three collaborating Bay Area high school teachers. BJC has been offered at UC Berkeley for three semesters, and 43 of the 77 college students in the 2009 pilot BJC at Berkeley chose to continue to the next, more demanding first course for CS majors. The UNC Charlotte pilot leverages PI Barnes' success with game development outreach and courses, and customizing Berkeley's 7 hour/week course to a 3 hour/week course.

Our team includes leading researchers and practitioners in education, broadening participation, and computer science education. We have an external evaluator with experience in the AP CS Principles, the former MIT Scratch developer who authored Snap, and an insightful multidisciplinary Advisory Board. Both the Snap software and the BJC curriculum (Moodle-based labs, lecture videos, and readings) are available online at no cost, and have already attracted worldwide interest. We are *already doing* all of the proposed activities on a smaller scale (for example, offering half-day teacher development workshops at conferences) but now we need support to move beyond what has so far been largely a volunteer effort.

We are inspired and informed by the STARS Alliance and leverage it to support our project. The STARS Alliance is a community of regional partnerships among academia, K-12 schools, professional and community groups catalyzed by the STARS Leadership Corps (SLC), a leadership program that engages college students in computing outreach, research, and service to broaden participation in computing. We propose to support regional partnerships of high school teachers with university professors and students to achieve the FRABJIOUS CS approach, augmenting STARS with new members and supporting this FCS project with the vibrant STARS community and annual STARS Celebration conference.

1. Project Goals and Outcomes

We will support the CS10K effort to broaden participation by creating a sustainable mechanism for teacher preparation to replicate our successful and engaging *Beauty and Joy of Computing* course. We first outline the course and the key computing concepts it addresses, and then list the specific goals and outcomes needed to bring the course to a wider, more diverse audience.

1a. The *Beauty and Joy of Computing* Course

The BJC course combines Moodle-based computer programming labs, lectures ranging from artificial intelligence and parallelism to the social implications of computing and technology, and small discussion sections. At Berkeley, each week of the 14-week semester includes four hours of lab, two of lecture, and one of discussion.¹ At UNC Charlotte, the three-credit hour course includes two 75-minute classes, divided into lab and discussion/lecture sessions each class. We anticipate that high school schedules will be different, but maintaining the half-programming ratio. Students work in pairs on midterm and final programming projects of their own choice, submit a term paper on a topic of their choice, and take midterm and final exams that span the programming and non-programming topics. The programming content features recursion and higher order functions, because of the powerful and beautiful way that they exemplify *abstraction*, one of the core ideas in computer science and in the AP CS Principles course design. UNC Charlotte adds modules on GameMaker, AppInventor, and StarLogo TNG, showing by comparison that Snap is not just a friendly programming language but a powerful tool that can be adapted to other contexts. In the non-programming part of the course we try to balance a fundamental optimism about the future of computer technology with an understanding of its limitations and potential for harm. Student readings are taken mainly from the excellent textbook *Blown to Bits* [Abelson 2008] supplemented with online articles and videos on special topics and current events.

Core Computing Concepts

The BJC curriculum introduces traditional introductory programming topics (e.g., variables, conditionals, loops, arrays), but also includes *recursion* and *higher order functions*, making the course more rigorous, and includes *an emphasis on the impact of computing on people*, making the course more meaningful and relevant to students. We emphasize the two programming techniques because they have an almost

¹<http://inst.eecs.berkeley.edu/~cs10/sp11/#calendar>

magical power to produce complex results from small programs, and because they exemplify *abstraction*, a central idea of computer science. We emphasize reflection on the impacts of computing on society because we want students to be *critical thinkers*, understanding the (often unintended) social, ethical, and economic implications of technology and computing. In each implementation of BJC, current events and interdisciplinary applications will be emphasized to connect with students and maintain relevance. For example, the UNCC course includes games, mobile device programming, and social networks.

Functions as Data and their Visual Representation

Many CS educators consider “functions as data” too hard for an introductory course. It is *not* a goal of this project to insist that everyone must teach functions as data. But it *is* one of our research goals to find out whether the inclusion of this topic improves the overall learning for participating students. A *higher order function* is one whose domain or range includes other functions. [Steele 1975] [Abelson 1996] Higher order functions are an extremely powerful means to capture patterns of control flow in a program, eliminating the need to keep track of index variables and individual array elements. For example, a conventional program to multiply each element of a list by 10 is given on the left below. To add 12 to each element instead, the user would have to write a separate procedure with the same logic. By contrast, the higher order function **map**, shown on the right, takes any function and a list as arguments, and without further programming, applies the function to each element of the list, and returns a new results list:



Students have traditionally found it difficult to understand and apply the use of functions as data. In Snap, a function block can be reified (used as a value) by using it inside a “the _ block” block:



The grey border around a block indicates that it is being used as a value, rather than as a recipe. We hypothesize that this explicit visual representation of a block-as-thing, and the use of empty input slots to avoid explicit formal parameters, make higher order functions accessible to all BJC students in high school and college. This hypothesis is supported by anecdotal evidence from our first group of Berkeley BJC students, but needs scientific study with a larger group of students. Although not all teachers may adopt this topic, we will include it in the teacher preparation workshops for all participating teachers.

1b. Goals

- In support of the **AP CS Principles initiative**, refine our *Beauty and Joy of Computers* curriculum design to attract and retain female and underrepresented minority students.
 - The initial offering of BJC at Berkeley was successful at attracting female and URM students, and in the achievement of female students, but not in the achievement of URM students. A vital research goal is to understand and correct the latter in future semesters.

- Closely connected to improving the student experience of our curriculum is improving the underlying software (Snap), especially its speed, its support for debugging, and making it available in the browser. Running in the browser will lift one of the major barriers to adopting new software at many high schools.
- In support of the **CS10K initiative**, develop an in-service teacher preparation program that will develop 100 diverse participant teachers' expertise in using the curriculum, and produce a second generation of teacher trainers.
 - Develop a teacher preparation model that involves each participant for two summers, with support for teaching BJC during the intervening year.
 - To provide for scalability, develop a model for the recruitment and support of additional university sites, and additional trainers from among previous cohorts.
 - Develop teacher support materials (e.g., Teachers' Manual, web site) for the course.
- Disseminate the results of our work through journal publication and presentation at conferences. All of our curriculum materials and software are available free online (csprinciples.org) and the results of the funded work will be also.

1c. Outcomes

Outreach and Professional Development Outcomes

- Offer BJC to over 2,000 students across at least 8 school districts. We will recruit and support these 8 districts through university-HS partnerships in regions including: Berkeley (UCB), Charlotte (UNCC), Durham (Duke), Austin, TX (UT-Austin), Los Angeles (UCI), and Chicago (TBD). We will recruit other regions through the CS Principles project and select those with large potential URM populations, interested teachers, and university faculty willing to form long-lasting relationships with high schools.
- Train 100 teachers in our workshops, including those without prior CS background. Teachers will learn the necessary programming skills and breadth of knowledge to teach the material effectively, as well as pedagogical approaches that support recruitment and retention of females and underrepresented minorities in computing courses.
- Develop a model of sustainable professional development, through professors and experienced teachers continuing to support novice instructors after the project end date, and through the definition of the necessary requirements for our professional development model to operate in new regions.

Assessing Curricular Effectiveness Outcomes

- We believe that asking high school students to rigorously explore recursion and higher order functions will uphold our goal of supporting female and URM students. We will widen this research question to include all of the BJC units, using models of successive curricular refinement to compare student learning and beliefs through embedded assessments, exams, and surveys [Linn, in press].
- In comparisons with the courses that are replaced by BJC, we expect that recruitment, retention, and enjoyment for female and underrepresented minority students to increase. We will also compare enrollment and course grades in computing courses into which BJC students continue, expecting that BJC will do as well or better than the courses it replaces.
- In support of efforts to promote the AP Principles course, we will evaluate whether and how successfully BJC is used in the universities in our university-HS partnerships. We will collect teacher and student outcomes.

Tools Development Outcomes

- Snap will be improved through speedup, better debugging support, support for running in the browser, and developing primitives for disk files and web pages as first class data.

1d. Impact and Sustainability

We will prepare **100 high school teachers**, including 16 teacher-leaders, emphasizing teachers of URM students, to teach the BJC course to over **2000 high school students annually** in at least 8 school districts. (The schools in cohort 0 range from 2% to 85% Black, and from 2% to 20% Latino.) These teachers will support one another in regional clusters with a university to form up to eight **new STARS Alliance locations** across the US supported by 16 additional STARS SLC college students. BJC will be taught at these partner universities, resulting in up to **8 new university adoptions** of the BJC course (some may already be participating in the AP CS Principles Pilot II), with at least 20 students per adoption, with the potential to reach at least **160-800 college students annually**.

Using the information gleaned from our education research component, we will refine the course, studying each unit for its appeal to female and URM students and for project teachers' success in teaching the unit. These results will contribute to the literature about **effective computing education practices** for broadening participation to women and URM students. The result will be a **freely available online course** that will have impact beyond the time period of the funding. Use of our BJC curriculum is already growing internationally, and this project will only accelerate the trend. With its modular construction, we expect that in the long run portions of the curriculum will remain relevant while other portions might have to change as the technology and its social context change. Similarly, Snap is open-sourced and, with the growth of users expected in this project and through the general growth of Scratch, we expect that it will be both relevant and under community development for some time.

Perhaps most importantly, if we are successful, FRABJOUS CS will represent a **scalable**, practical, and sustainable model for decentralized **teacher preparation** that leverages the successful STARS Alliance to support high school teachers and strengthen partnerships between K12 and universities. By gradually reducing support and measuring effectiveness, we will gain understanding what the financial and staffing requirements are for successful professional development. While good professional development will never be free, with a reasonable enough cost structure additional funding can be sought or the services can be performed on a fee-charged basis. Since we are not only training teachers but also training trainers and establishing independent clusters elsewhere, sustainability will not depend just on the PIs of this project. We hope to have a viral effect, where leaders at each FRABJOUS CS site will continue the process of spreading BJC and our approach to additional sites.

If our research hypothesis proves correct, we will set a high standard for the **technical rigor** of a course to which female and URM students will be attracted, and in which they will succeed. We expect this high standard to be reflected in the eventual AP exam. With universities adopting versions of BJC, we believe that other university instructors will become as inspired, as we have been, to **inject passion, beauty, joy, and awe back into computing curricula** beyond this course, and students will demand that computing courses promote collaboration, creativity, critical thinking, and cultural, social, and ethical awareness along with rigor and technique.

1e. Relevant Research and Prior Work

High school computer science education in the United States has shown significant declines in both the number of introductory CS courses being taught [CSTA 2010], and the intention of students (especially those from underrepresented populations) declaring computing as a major [HERI 2009]. To address this, the College Board is exploring the creation of a new, engaging advanced placement course to teach computational thinking [Wing 2006] and broaden participation [Astrachan 2010], based on prior efforts to define "core principles and key concepts" [CSTA 2006]. We were fortunate to have been chosen as two of the five first-round national pilots during the academic year 2010-2011 [Astrachan 2011]. The CS10K project hopes to have 10,000 teachers in 10,000 high schools teaching this new AP course by 2015 [Astrachan/Cuny 2011].

We are both informed and inspired by the outstanding outreach work of Margolis et al to underrepresented populations [Margolis 2003], [Margolis 2008]. Their recent *Exploring Computer Science* curriculum was explicitly designed for broadening participation, to channel the interests of urban HS

students with “culturally relevant and meaningful curriculum” [Goode 2011]. In moving BJC to high schools, we hope to model some of their best practices for our curricular refinement and PD component, as well as some of the “messages” for young people that have been shown to be effective [WGBH 2009]. While working with underrepresented high school students, we draw from our decade-long success with the BFOIT summer institutes, which have served over one hundred fifty students, sixty-five who have gone on to attend some of the top universities in the country [Crutchfield 2011].

Joyce and Showers [1980, 2002] categorized the different types of in-service training into a hierarchy, and highlighted the importance of “theory, demonstration, practice, feedback and coaching” to achieve the most successful transfer to the classroom. We have learned from this and plan to include peer-peer and yearly online support, as well as monthly day-long face-to-face training and check-in sessions to sustain our learning community. We will draw from our experiences with successful high school teacher outreach, through the IFSMACSE program [Harvey 1992], the summer 2010 CS4HS workshops at UCB [CS4HS 2010]. We also understand the critical importance of solid evaluation in assessing effectiveness of these interventions [Davis 1985].

Our Snap software [Snap 2011] that underlies the programming component of our BJC curriculum is based on Scratch, created to teach “designing, creating and remixing” to today’s “digital natives” [Resnick 2008]. Some have explored learning CS concepts with it [Meerbaum-Salant 2010], but the lack of named functions [Maloney 2010] meant that we couldn’t use it to teach recursion, a core computing topic in our non-majors class at UCB. Co-PI Harvey and developer Jens Mönig worked to add a few key features to Scratch to serve both kids and computer scientists and share their free software with others [Harvey 2010a, 2010b]. Finally, in our closed labs, our students use a lab-centric instructional model that provides several benefits to students, namely “frequent and varied self assessments, and integrated collaborative activities” [Titterton 2010].

2. Implementation Plan: A Cycle of Innovation and Learning

PI Garcia and Co-PI Harvey began studying the question of a new approach to CS education for non-majors in 2008-09. Berkeley’s previous course for non-majors (CS 3, *Introduction to Symbolic Programming*) had originally been designed to try to serve the needs of two very different populations simultaneously: business students, who were required to take a programming course but generally had no real interest in it, and would-be CS majors who hadn’t programmed in high school and were therefore not ready for our first course for majors. By 2008, the business requirement no longer existed, and many of our CS majors viewed CS 3 as a de facto requirement even if they did have programming experience. The result was that CS 3 was no longer well suited to the needs of curious non-majors. It was at about the same time that the College Board began the design of the AP CS Principles course, and we quickly saw the connection between that project and our own internal needs. **Both of our campuses are among the five initial pilot sites for AP CS Principles;** the PIs of this proposal are the developers and instructors of those pilot courses.

At first we thought that the new course we were designing would be for non-engineers, while CS 3 would continue as the intro course for engineers. We designed about half of the course content in spring, 2009, and offered this half-course to a selected group of non-engineering students in fall, 2009. These pilot students were surveyed before and after the course, and the results of this evaluation informed the second cycle of course development in 2009-2010. At the same time, we were working with Jens Mönig, who had added a user-defined block (procedure) capability to an extended version of Scratch, to add the further capability of first-class procedures and, therefore, higher order functions. That development made it possible for BJC to meet the technical needs even of engineering students.

The first full offering of our UCB BJC class (CS 10) was in fall 2010; the second is currently in progress. This semester, more than half of the students from last semester’s CS 10 have chosen to take our subsequent first course for CS majors (CS 61A), and we are beginning to collect data about their success in that more demanding course. (As we write this, 11 weeks into the 14-week semester, retention and performance of the students who took CS 10 before CS 61A are about the same as that of students whose preparation for 61A happened elsewhere.)

Instead of developing a new CS Principles course from scratch, PI Barnes observed the other 3 pilot sites during fall 2010, concluding that UCB's BJC course best fit UNCC's needs for a rigorous introductory computing course that could highlight exciting opportunities for students. UNCC has unique opportunities for research with faculty, and has even offered these opportunities to high school students in the area, particularly in the summer. Dr. Barnes started teaching game design and development (GDD) at UNCC in 2005, and developed a GDD certificate program first offered in 2008. Her research in serious games has been around making educational and other games for a purpose, and she offered a very successful Serious Games course in Fall 2009 [Chaffin 2010]. Central to her games courses has been a student-chosen project, with an emphasis on rapid prototyping with frequent feedback. Therefore, when approaching the CS Principles pilot course at UNCC, PI Barnes chose to adopt UCB's framework, but augment it with labs using GameMaker and AppInventor, and extending the project to include playtesting and revision. UNCC is also extending the simulation module to include StarLogoTNG, a free simulation tool. Although Snap is an ideal language for a first computing course, PI Barnes strongly feels that exposure to several low-threshold languages helps students better understand object-oriented principles and the commonalities of programming languages and environments without explicit instruction.

With this proposal, we are seeking funding for a full-scale evaluation of the curriculum, both on our campuses and in high schools. We have begun collaborating with high school teachers and will carry out teacher preparation for more teachers starting this summer (2011) with other funding. This proposal will fund further support for those teachers during the 2011-12 school year and engage these teachers as teacher-leaders in Year 1 to train regional cohorts of teachers to adopt the BJC course. Each year we will conduct formative evaluation to refine our professional development and curriculum to make it more effective, with a particular focus on women and underrepresented minority teachers and students.

2a. Outreach and Professional Development

We have already built relationships with 13 in-service high school teachers in California and four in Charlotte, NC who plan to teach some or all of our curriculum in the 2011-2012 academic year. We are planning a teacher preparation workshop this (2011) summer, using non-CE21 funds, for participant teachers at each of our two campuses. Participants have been recruited primarily through our connection with the Golden Gate CSTA in California and the Charlotte-Mecklenburg schools in North Carolina. To support professional development workshops, we will collaboratively work with this first cohort of teachers to develop a Teachers' Manual, an index of correlation between the weekly BJC units and the AP CS Principles documents, and a collaborative working site where teachers can discuss the curriculum, create and share their new modules and modifications of existing modules, and fully plan their own courses for the upcoming year. This will also include online forums that will keep participants updated about monthly PD meetings and project announcements.

The workshop will work on a 1- n -1 model: an initial in-person meeting for one week to introduce the curriculum and software, followed by n weeks in which participants take the course online from home, and a final one-week in-person meeting to work together to form conclusions and prepare for teaching. (The value of n will depend on local conditions and school calendars; Berkeley will be offering the BJC course in our 8-week summer session starting in 2012, and one possibility is that teacher participants will take the course online, so $n=8$, with the option of earning Berkeley continuing education units.)

We seek funding to support continued work with these participants during the 2011-12 school year, including monthly local meetings of neighboring subsets, email and telephone consultation with faculty, and refinement of the curriculum in response to classroom experience. The monthly meetings will be entirely about the teachers' experience in teaching the course during the early months, but will also include some presentation of CS topics beyond this curriculum (e.g., automata theory) in the later months so that the teachers have a larger context for their work. We will repeat this model in Year 1 of this proposed CE21 grant, with modifications based on what we learn in summer 2011 (Year 0). Our plan is to bring Year 0 teachers back in the summer of Year 1 to become teacher-leaders who will help conduct the training for the next cohort of teachers after a brief training just before the second cohort arrives. We will also work to engage university professors at STARS Alliance institutions to teach a parallel course and develop regional clusters of teachers who can be similarly supported as we are doing in Charlotte

and Berkeley, with local monthly meetings and low-cost summer workshops with limited travel and lodging costs. We anticipate engaging two other university professors and high school teacher cohorts in Year 1, and three other clusters in each of Years 2 and 3. **100 computer science teachers will be prepared, contributing to the CS10K initiative.** We plan to submit a Type II proposal to expand the impact of this tiered training model at a larger scale, and reaching out beyond existing CS teachers through publicity in general teaching organizations such as NEA and AFT, as soon as we find our method is effective.

We will use the **ensemble** NSDL [Fox 2010] computing portal to allow teachers who download and adopt our material to provide feedback about what was excellent and what was a dud at the individual item level. It will allow commenting, rating and tagging, as well as support our emerging learning community through forums and blogs.

The format of preparation over two consecutive summers with support activities during the intervening year comes from the extremely successful Institute For Secondary Mathematics And Computer Science Education (IFSMA-CSE), an NSF-sponsored in-service teacher preparation program held at Kent State University, Ohio, in 1989-92, in which Co-PI Harvey was an Associate Director. Several alumni of that program did go on to be leaders of teacher preparation workshops for Ohio teachers.

We hope to scale our efforts through a tiered and regional approach to building a community and momentum around BJC and CS10K. The first cohort of teachers we will train this summer (under separate funding) includes several talented and engaged teachers who are already working with us to build the local communities of teachers, and are willing to travel to help conduct future PD workshops as mentor teacher participants in this project. Eugene Lemon, a computer science teacher at the Ralph J. Bunche High School in Oakland, CA, and Sharon Jones, a computing teacher at Phillip O'Berry Academy in Charlotte, NC who just received her PhD in Education, have already begun to explore our curriculum, will participate in the Year 0 (unfunded) summer workshop, and will teach BJC during 2011-12. We plan to support mentor teachers like these two exemplary teachers through NSF Research Experiences for Teachers, as we will work with these teachers to build their leadership and teaching skills while they also help evaluate the BJC curriculum.

2b. Assessing Curricular Effectiveness

The main thrust of our educational research centers on determining the effectiveness of BJC in engaging and educating females and URM students. The initial Fall 2010 offering of BJC at Berkeley was extremely successful with female students (45% of the class), with this enrollment rate equal to the 11-year high in the previous (programming-only) version of the non-majors course [Lewis 2010]. These 34 young women also performed much better than their male counterparts, with an average grade of 3.06 (a little over a B) vs 2.87 (a little over a B-) for the men. Finally, the top student was a woman, for the first time in six semesters that PI Garcia had taught an introductory course.

A larger proportion (21%) of BJC students were from underrepresented minority (URM) groups than typical introductory computing courses at Berkeley, but they had a lower average course grade (2.0) than majority students (3.25). One research focus of this project will be to determine reasons for this result and improve the curriculum to better support, scaffold and engage URM students. At UNC Charlotte, the much smaller course of 22 students has four African American males, and four women. The course will be offered again in the fall, we hope with broader participation across women and URM students.

There is a wide array of factors that affect access and success in computing education, including well-researched factors such as opportunity [Margolis 2008], relevance [Varma 2009, Barker 2009], and classroom climate [Garvin, 2004]. We believe that both relevance and classroom climates are strong mediators for the success we've observed with the BJC curriculum. However, as part of this project we will investigate the particular BJC curricular elements that may contribute to differential success with women and URM students. As such, we hope to be able to concretely support our curricular refinement process and end with a course that has consistent and effective modules.

As our project title suggests, we believe that technically rigorous computing concepts are not at odds with engagement and access for *any* student, when framed with our tools and supporting curriculum. Similarly, Goode [2006] provides evidence that rigorous math-based computing curriculum can engage females. Of particular interest in BJC are the modules on higher-order functions, recursion, distributed computing, concurrency, and simulation, but also those concerning basic programming techniques, game design, the social implications of computing, and artificial intelligence. We will employ a model of successive refinement across offerings and measure growth in our relevant outcomes, similar to that used by Linn [2009, in press] in high-school science. Our outcomes will include module-specific knowledge, as measured by embedded assessments and exam questions, survey items on engagement, and mining of access data collected by the learning management software.

In addition to the main research question on curricular modules, we will examine the effect of BJC on recruitment and retention of females and URMs. In this effort, we will encourage our sites to collect retention and grade data with gender and URM indicators for courses that will be replaced by BJC. Some of this may require data collection by the participating instructors, while other institutions collect the data automatically. Additionally, collection of enrollment and grade data for courses that are expected to follow BJC will be encouraged, although not applicable for cohort 3 within the project timeline.

Finally, we will examine how the BJC course has impacted instruction at the university level in our 8 sites. By encouraging and supporting this, we hope to support approval of the AP Principles course, which is likely to act as a driver in spreading BJC to high schools.

2c. Tools Development

As part of this project, Jens Mönig, the (volunteer) developer of BYOB/Snap, will collaborate with us as a consultant programmer to improve the speed of running users' programs, add support for debugging, and support running Snap in a browser. All of these improvements will make Snap dramatically more usable while also making it easier to deploy in schools where installing new software is a challenge. We will continue to apply the cycle of learning and development to inform Snap development. For example, BYOB 3.1 (rel. 4/2011) supports object oriented programming of sprites (the animated Scratch characters) with true inheritance, enabling the addition of prototyping OOP [Lieberman 1986] to the curriculum.

2d. Dissemination

All of our curriculum materials (Moodle lab units, lecture videos, lecture notes/slides, and overall course descriptions and calendars) are available online, both at our university sites and at csprinciples.org, through a *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported (CC-BY-NC-SA)* license. The complete *Blown to Bits* textbook is also freely available online, as are the Snap programming language and the Moodle lab administration software. We shall continue to make course and software revisions available online at no cost.

The University of California Office of the President (UCOP) is funding a *UC Online Instructional Pilot Project* to assess whether "online instruction can use technology's tools to give undergraduates educational opportunities comparable to the superb classroom instruction that helped build UC's stellar reputation worldwide" [UCOP 2011]. UCB's BJC course has been chosen as one of the first to be offered in this online form, and we are excited about the possibility of offering a normal, 8-week, summer 2012 version of the course to local students, with a synchronous online cohort taking the course through the Online program. Remote participants would watch lecture videos, work on labs and Snap projects, and be provided Internet-mediated support and video-conference discussion from teaching assistants. We anticipate that this project will bring worldwide attention to our curriculum, and could truly help the CS10K initiative, since there is an almost endless supply of student teaching talent who would be willing to work as teaching assistants for the summer. When we describe the "1-N-1" model of summer PD, the "N" weeks of remote learning sandwiched between the 1-week face-to-face sessions could be this 8-week block to take the UCB BJC Online course remotely.

We have already presented our curriculum at SIGCSE conference sessions and workshops. The Snap programming language has been presented at the Connectionism 2010 conference, at the 2010 MIT Scratch conference, and will be presented at this summer's CSTA CS&IT and STARS Celebration conferences. We shall continue to present our work at similar conferences and submit our research results to refereed journals on computer science education.

2e. Timeline

We propose a three year project. Snap development is concentrated in the first year. We balance research and outreach by conducting teacher training and support each year but focus targeted efforts to understand the impact on students at a few select high school courses with high proportions of URMs. We will study teachers trained directly by us in the first year, and teachers trained by our trainees (thereby evaluating the scalability of the project) in the second year. Revision and improvement of the curriculum in response to the research results will continue throughout the project but is considered part of our teaching duties and requires no NSF funding.

2011 pre-project	Summer: PI-led Cohort 0 summer PD workshop, separately at UNCC & UCB Summer: STARS Celebration BJC workshop and professor recruiting Fall: Cohort 0 High-school BJC courses & monthly teacher meetings begin
Every semester	Advisory Board meeting Pilot use of new tested Snap features, refine BJC course
Every fall	High School BJC courses & monthly participant meetings begin Recruit next cohort sites and participants
Spring 2012	Snap development begins Post evaluation of Cohort 0 BJC courses
Summer 2012	PI & mentor teacher-led Cohort 1 PD workshops (20 with stipends) in Durham, NC & LA Kevin Wang brings BJC curriculum to Seattle TEALS
Summer 2013	Cohort 2 teachers (40 estimated) attend workshops at 4 sites
Summer 2014	Cohort 3 teachers (40 estimated) attend workshops at 4 sites

3. Evaluation Plan

Evaluation is a critical component of FRABJIOUS CS, to assess both the effectiveness of the BJC curriculum and scaled, teacher-led professional development and to motivate other institutions to initiate similar actions. Both formative and summative evaluation measures will be used to inform and refine activities through the course of the project, and to determine the overall success of the project in reaching its goals and outcomes. The overall evaluation questions, planned instruments, and evidence we anticipate for project activities are below.

Evaluation Questions	Potential Instruments	Potential Evidence
<p>Outreach & Professional Development:</p> <p>-To what extent are recruitment goals met for numbers and proportions of diverse teachers&students?</p> <p>-To what extent is the teacher PD conducted as planned (e.g., B hours of PD)?</p> <p>-To what extent do teachers feel prepared to teach the course?</p>	<p>Documentation of recruitment effort</p> <p>Documentation and observation of PD process</p> <p>Participant survey for PD</p>	<p>Description of recruitment results, with stakeholder sign-off</p> <p>e.g., PD training agenda indicating hours and topics</p> <p>Participant outcomes from PD, including:</p> <ul style="list-style-type: none"> - level of comfort with curricular materials - felt confidence and preparedness for teaching course, including use of Snap, course topics, class activities
<p>Assessing Student Learning:</p> <p>-[Efficacy of implementation] To what extent do students gain in computational thinking skills and course topic knowledge?</p> <p>-Are there differences in computational thinking skills across student groups?</p> <p>-What implementation factors are associated with gains in student learning?</p> <p>-What other factors are associated with gains in student learning for diverse students?</p>	<p>Assessment- computational thinking skills and course topic knowledge (developed by collaborating teachers)</p> <p>Ongoing teacher survey of course implementation (e.g., what instructional approach, what topics/activities, in what depth, how often?)</p> <p>Student survey of background (e.g., course-taking, technology experiences) and non-cognitive (e.g., interest, self-efficacy, identity, confidence) factors</p>	<p>Class and student-level results based on pre/post admin's of assessment and survey for all students (fall 2011, spring 2012), and mid-year think-aloud study for student subset:</p> <ul style="list-style-type: none"> -initial CT skills and course topic knowledge -gains in CT skills, course topic knowledge -specific use of CT skills in problem-solving -implementation factors related to learning gains (e.g., teaching recursion, higher order functions, game design) -other factors related to learning gains
<p>Curriculum Development:</p> <p>-To what extent do curriculum materials meet development goals (e.g., expected examples)?</p> <p>-To what extent are materials likely to support the success of under-represented minority students?</p> <p>-Are evaluation results effectively translated into subsequent curriculum revisions?</p>	<p>Advisor curriculum review protocol</p>	<p>Alignment of materials to goals, strengths and weaknesses of materials</p>
<p>Tools Development:</p> <p>- To what extent are the improvements to Snap achieved (including speed, debugging support, and running in browser)?</p>	<p>Interviews with tools developer</p> <p>Student survey (see above)</p>	<p>Description from tools developer about specific development work addressing these goals</p> <p>Student experiences and satisfaction with speed of Snap and debugging functionality</p>

4. Partnership Plan

Both UC Berkeley and UNC Charlotte have attested that they will provide course credit or placement for a course like BJC if offered as an AP course. UC Berkeley has demonstrated their commitment to this project by incorporating BJC as its introductory computing course for non-majors, with about 100 students taking it each semester. UNC Charlotte is the only university in NC with a College of Computing and Informatics, and as the lead institution in the STARS Alliance has already build a significant infrastructure to support community outreach both locally and nationally. UC Berkeley and UNC Charlotte have already piloted a small scale model of what we propose here: to replicate the BJC curriculum through universities and establish these universities as hubs for further replication, just as the STARS Alliance builds a community to broaden participation centered at universities. Therefore, we propose to recruit promising groups to replicate our model and integrate these groups into the STARS community. The partnership between UC Berkeley and UNC Charlotte will be achieved through communication via a listserv mailing list, monthly Skype meetings and two annual meetings of all key personnel. The partnership for the FRABJOUS CS project with the STARS Alliance will be aligned with existing hierarchical Alliance structures. Each Alliance partner university is coordinated through an Academic Liaison, who communicates with the Alliance, and directs college students in their STARS Leadership Corps (SLC) projects to broaden participation through computing outreach, research, or service. Each year, STARS faculty and students convene at the STARS Celebration to build community, present their work, and conduct training for SLC projects. STARS Alliance-wide evaluation is conducted by the STARS evaluation team, with local students helping to collect data. Roles of the key personnel are listed in Table 1.

Table 1. **FRABJOUS CS Roles and Partnership Plans**

Role	Description
PIs	<u>Dan Garcia</u> , <u>Brian Harvey</u> , and <u>Tiffany Barnes</u> will direct the project, and will be responsible for coordinating all training, management, web deployment and NSF reporting. PI <u>Garcia</u> will lead mentor teachers, including Eugene Lemon, to create the PD curriculum, and collaborate with the Golden Gate CSTA to support teachers. PI <u>Barnes</u> will help form a Charlotte CSTA chapter and will formalize short-format BJC curriculum.
Snap Dev.	Co-PI <u>Harvey</u> will be responsible for day-to-day communication with lead software developer <u>Jens Mönig</u> , located in Germany, by Skype and email. This is a continuation of the two-year collaboration that led to the development of BYOB 3.0 and 3.1.
Project Manager & Ed. Research Lead	As Project Manager, <u>Nathaniel Titterton</u> , Berkeley, will direct project communication and maintain the project timeline. Dr. Titterton will work with external evaluator Kathleen Haynie to ensure that formative and summative evaluation are conducted. He will also lead the education research component of the project, to more deeply investigate the effectiveness of our curricular modules with URM students.
External evaluators	Advisory Board Chair <u>Marcia Linn</u> will be responsible for advising the project team on the development, deployment, and evaluation of the professional development for in-service teachers. <u>Kathleen Haynie</u> , evaluation consultant, will be responsible for development, dissemination, and collection, and analysis of evaluation instruments to be used specifically with BJs. She will work with Barnes, Linn, and Titterton as needed for data collection and coordination.
STARS-BJC Director	PI <u>Tiffany Barnes</u> , UNC Charlotte, is co-PI of the STARS Alliance will be responsible for interfacing between BJC and STARS, as well as recruiting and coordinating faculty mentors of SLC students supporting BJC. She will ensure that each regional location has recruited high school teachers to teach BJC and that BJC information is disseminated through StarsAlliance.org.
STARS Academic FCS Liaisons	A university faculty member adopts the BJC course at the university level, mentoring two STARS SLC college students to help with the course and supporting a local community of high school teachers in a cluster through forming and strengthening CSTA chapters. These do not have to come from existing STARS institutions; locations will be selected and funded based on potential impact by number of students overall and in URMs. Dr. Barnes will work with each of these liaisons to coordinate FCS efforts at each institution.

4a. Project Management Team

Principal Investigators

Dan Garcia (PI, UCB) is a Lecturer with Security of Employment in the EECS Department at the University of California, Berkeley. He received his Ph.D. in Computer Science from UC Berkeley in 2000. He was the co-developer and co-instructor (with Brian Harvey) of Berkeley's Advanced Placement *Computer Science: Principles* pilot course *CS 10: The Beauty and Joy of Computing*. He is a member of the Advisory Board of the *Berkeley Foundation for Opportunities in Information Technology* (BFOIT), a free, year-long CS outreach program for fifty 6-12 grade underrepresented students that culminates in a two-week summer institute, and a shorter program for ten 3-5 graders. He is also on the Advanced Placement *Computer Science: Principles* Advisory Board, and the ACM Education Board. He won all four (as a graduate student and Lecturer) of the computer science division's outstanding teaching awards, and holds the record for the highest teaching rating in an introductory course in the history of the division.

Brian Harvey (co-PI, UCB) is a Lecturer with Security of Employment in the EECS Department at the University of California, Berkeley. He received his Ph.D. in Science and Mathematics Education at Berkeley in 1985. He has been a high school computer science teacher and currently volunteers in an elementary school. He is co-author of *Simply Scheme: Introducing Computer Science*, the textbook for our former CS0-level course, and is the author of the three-volume *Computer Science Logo Style* books for teenagers. He is co-developer of the Snap programming language. He developed and taught the first summer course at the Berkeley Foundation for Information Technology (BFOIT) and currently teaches the BFOIT program for elementary school students using Scratch and is a member of its Advisory Board. He was an Associate Director of the NSF-sponsored IFSMACSE in-service teacher development program for Ohio math and computer science teachers (1989–92). He has won the prestigious Berkeley campus-wide Distinguished Teaching Award as well as two EECS Department teaching awards.

Tiffany Barnes (PI, UNCC) is an Associate Professor of Computer Science at the University of North Carolina at Charlotte and received her PhD in CS at NC State in 2003. Dr. Barnes is co-PI on the NSF-BPC funded STARS Alliance grant that engages college students in outreach, research, and service [cite STARS]. She is PI on the NSF-BPC funded Culturally Situated Design Tools to expand the creation of tools to simulate cultural design with software through the STARS Alliance, and this work on cultural games and learning tools helps computing and mathematics appeal to a broader audience. Dr. Barnes is Director of the NSF-funded Game2Learn Research project that engages students in developing and testing games to teach introductory computing, with the goal of broadening participation and increasing learning. Dr. Barnes received an NSF CAREER Award to develop new technologies that leverage data to adapt STEM learning software to individual students and make this data more understandable for teachers. She has had extensive outreach experience. She has been program Technical Director (1998-2006) for "Girls on Track," an NSF-funded summer program designed to increase girls' enthusiasm for and confidence in learning mathematics and using computer technology [cite GOT]. She and her team have taught game development and culturally situated design tools summer camps for middle, high school and college students and afterschool activities for middle schoolers each year since 2005.

Other Senior Personnel

Nathaniel Titterton (project manager and educational research lead) is a research specialist in the EECS Department at U.C. Berkeley, receiving his PhD at UC Berkeley in Education. He has researched lab-centric pedagogy in computing education [Titterton 2010] and been project and educational research lead on several NSF- and State-of-California-supported course development projects.

Marcia Linn (Advisory Board chair) is professor of development and cognition in the Graduate School of Education at U.C. Berkeley. She is a member of the National Academy of Education and a Fellow of the American Association for the Advancement of Science, the American Psychological Association, and the Association for Psychological Science. She has served as Chair of the AAAS Education Section and as President of the International Society of the Learning Sciences. Board service includes the American Association for the Advancement of Science board, the Graduate Record Examination Board of the Educational Testing Service, the McDonnell Foundation Cognitive Studies in Education Practice board, and the Education and Human Resources Directorate at the National Science Foundation. She has twice

been a fellow at the Center for Advanced Study in Behavioral Sciences. Linn has authored several books on teaching and technology.

Jens Mönig (software consultant) is the lead developer of Snap and a programmer and attorney with MioSoft Deutschland GmbH. He studied Law at the University of Tübingen and is a member of the Stuttgart bar. He has been a software consultant for the MIT Media Lab on the Scratch project.

Kathleen Haynie (external evaluator) is the Director of Haynie Research and Evaluation, specializing in preK-16 STEM evaluation. She received her Ph.D. in educational psychology from Stanford University in 2001. She has already served as the external evaluator for the AP CS Principles pilot 1. Her work on other NSF and NASA-funded projects includes the areas of science education (e.g., geomorphology, climate change, genetics), assessment design, technology use, and teacher professional development.

4b. 2011–2012 (year 0) BJC Pilot High School Teachers

We are delighted to have confirmed 17 teachers (13 clustered around UCB, 4 around UNCC) who, with the blessing of their principals and/or superintendents, will participate in our pre-project professional development and teach BJC in their high schools in the academic year 2011-2012. Below is a table listing the school or district's name and location, a summary of the participants and the demographics of the student body. We have indicated the teachers who have already been working with us this past year through the AP CS: *Principles* pilot initiative in **bold**.

School / District, Location	Teacher(s)	Demographics
Oakland (CA) Unified School District (Ralph Bunche, Oakland Tech, Skyline, and Castlemont)	Eugene Lemon , Emmanuel Onyeador, Sage Moore, Mark Frye, Nathan Burke.	AA: 29%; Latino: 36%; Only 60% graduate.
Encinal HS, Alameda, CA	Mary Clarke-Miller, Martha Lopez	AA: 20%; Latino: 20%; Filipino: 20%; Asian: 20%
Galileo Ac. Sci & Tech, SF, CA	Ben Chun	AA: 6%; Latino: 10%; ELL: 24%
Albany HS, Albany, CA	Ray Pedersen , Sean Morris	AA: 7%; Latino: 13%; Asian: 39%
Piedmont District, Piedmont, CA	Nathan Mattix, Jana Branisa	AA: 2%; Latino: 2%; Asian: 20%
Henry Gunn HS, Palo Alto, CA	Josh Paley	AA: 2%; Latino: 8%; Asian: 37%
Phillip O. Berry, Charlotte, NC	Sharon Jones, Beth Frierson	AA: 86%; Latino: 6%; Asian: 1.6%
Independence, Charlotte, NC	Brian Nivens	AA: 56%; Latino: 20%; Asian: 4%
N. Mecklenburg; Charlotte, NC	Renada Poteat	AA: 42%; Latino: 9%; Asian: 2%

4c. Advisory Board

Our Advisory Board will meet twice a year, and will assist in all project phases, with advisors providing direction in learning and engagement (education research), broadening participation, and CS.

Learning and engagement: **Marcia Linn** (Professor, School of Education, U.C. Berkeley) serves as the chair of the advisory board. She has authored seminal articles and several books in educational research areas such as educational technology, models of student learning, and gender equity in science education. **Mark Guzdial** (Professor, Georgia Institute of Technology) is a leader in computing education research, serving on the boards of prominent journals and groups in the field. **Michelle Hutton** (President of the Computer Science Teachers Association) has research focuses on middle and high school computer science education and gender equity in computer science, and is an author of the Exploring Computer Science curriculum. **Chris Stephenson** (Executive Director of the Computer Science Teachers Association) is the former chair of the ACM K-12 Task Force and a primary author of *Ensuring Exemplary Teaching in and Essential Discipline* (ACM, 2009). **Todd Ullah** (Principal, Washington Preparatory High School in Los Angeles) was Co-PI of the project that created the “Exploring Computer Science” curriculum. He has been Director of Instructional Technology Applications and Director of Secondary Science for the LA Unified School District.

Broadening Participation: **Jane Margolis** (Senior Researcher, Graduate School of Education and Information Studies, UCLA) has written trailblazing books on the broadening participation in computing (Margolis, Fisher, 2003; Margolis, Estrella, Goode, Holme, Nao, 2008). **Orpheus Crutchfield** (Executive Director, *Berkeley Foundation for Opportunities in Information Technology*) leads the decade-old foundation's work on exposing pre-college women and ethnic minorities to CS and engineering, preparing them for university-level study. **Sheila Humphreys** (Director of Diversity, EECS, UC Berkeley) has created mentoring programs both for women and underrepresented minority students to increase and sustain diversity, and authored books on evaluating gender equity (Davis and Humphreys, 1983). **Colleen Lewis** (Ph.D. candidate, Graduate School of Education, U.C. Berkeley) has conducted research in pre-college outreach, developed best practices for student learning and developing students' motivation to pursue computer science, and helped develop the BJC course.

Computer Science **Michael Clancy** (Senior Lecturer, EECS, U.C. Berkeley) is a leading CS Ed. researchers, with experience in self-paced instruction, misconceptions in computing education, case studies, pair programming, peer instruction, and lab-centric pedagogy. **Daniel Ingalls** (SAP Research), has helped create ground-breaking programming languages and environments, including Smalltalk, and Squeak EToys. **John Maloney** (researcher, MIT Media Lab) is the lead developer of the Scratch project. **Mitchel Resnick** (Professor, MIT Media Lab) has authored seminal papers on technology for computing education, and lead the research team that developed the Scratch programming environment.

5. Results from Prior Support

BPC-A & BPC-AE: Scaling the STARS Alliance: A National Community for Broadening Participation through Regional Partnerships (NSF-0540523, \$2,530,640, 3/1/06-2/28/10; NSF-0739216, \$3,009,870, 2/1/08-1/30/11; NSF-1042468, \$750,000, 1/1/11-12/31/16). Barnes is co-PI and evaluation coordinator on this project to broaden participation in computing. Over 800 students, mainly from underrepresented groups in 20 colleges/universities, participated in the STARS Leadership Corps (SLC) to perform computing-related outreach, research, and service since 2006, reaching over 25,000 in the K-12 community.

BPC-DP: Improving Minority Student Participation in the Computing Career Pipeline with Culturally Situated Design Tools (CSDTs), (NSF-0634342, \$219,109, 1/1/07-12/31/11). Barnes is PI on this grant to engage SLC students in outreach and to develop new tools that engage kids in learning computer science while they design culturally motivated artifacts, resulting in 3 STARS universities performing outreach through summer camps, after-school programs, and supporting undergraduate research to create new tools, including a dance programming tool and the Bead Loom Game.

Socially Relevant Computing Research for Undergraduates: Visualization, Virtual Environments, Gaming, and Networking (NSF-081745, \$393,561, 5/1/09-4/30/12). Barnes is co-PI for this REU Site that has supported over 50 undergrads; 90% of our graduating REU seniors are now in PhD programs.

CreativeIT: Pilot: Game2Learn: Creating Computing Education, (NSF-0757521, \$216,000, 6/1/08-5/31/12). Barnes is PI on this grant that formalized her successful Game2Learn project that engages advanced undergraduate students in building games that teach computing to first-year students into a new Serious Games class that leverages design challenges and processes for learning computing.

CAREER: Educational Data Mining for Student Support in Interactive Learning Environments (NSF-0845997, \$646,982, 7/1/09-6/30/14). Barnes is PI on this grant to develop new technologies that use data to adapt STEM learning software to individual students. She is developing a visual analytics tool to enable teachers and researchers to interactively model student solutions annotated with data that reflects frequency, tendency to commit future errors, and closeness to a final solution.

Institute for Secondary Mathematics and Computer Science Education (NSF-9055516, \$774,507, 3/15/91-8/31/92; NSF-8850509, \$1,691,457, 12/1/88-5/1/91). Harvey was Associate Director of this project that prepared 90 new high school computer science teachers as well as introducing 120 high school math teachers to the use of computer technology in mathematics learning.