

Creative Computation in High School

Dianna Xu
Computer Science
Bryn Mawr College
Bryn Mawr, PA (USA)
dxu@brynmawr.edu

Aaron Cadle
Computer Science
Fort Worth Country Day School
Forth Worth, TX (USA)
aaron.cadle@fwcd.com

Darby Thompson
Computer Science
Sidwell Friends School
Washington, DC (USA)
thompsond@sidwell.edu

Ursula Wolz
RiverSound Solutions
Montclair, NJ (USA)
<http://riversoundsolutions.com>
riversoundsol@gmail.com

Ira Greenberg
Computer Science & Engineering
Southern Methodist University
Dallas, TX (USA)
igreenberg@smu.edu

Deepak Kumar
Computer Science
Bryn Mawr College
Bryn Mawr, PA (USA)
dkumar@brynmawr.edu

ABSTRACT

In this paper we describe the success of bringing Creative Computation via Processing into two very different high schools that span the range of possibilities of grades 9-12 in American education. Creative Computation is an emerging discipline that requires a thorough grounding in both media arts and computing. We report on how contextualized computing that supports integration of media arts, design, and computer science can successfully attract and motivate students to learn foundations of programming and come back for more. The work of two high school teachers with divergent pedagogical styles is presented. They successfully adapted a college-level Creative Computation curriculum to their individual school cultures providing a catalyst for significant increases in total enrollment as well as female participation in high school computer science.

CCS Concepts

• Social and professional topics~K-12 education

Keywords

K-12; CS0; CS1; education; creative computation; Processing.

1. INTRODUCTION

Contextualized computing provides a means through which a diverse student population gains insight into the foundations of computer science [Cassel & Wolz 2013]. At the college level this includes media computation [Guzdial 2004], robots [Summet et al 2009, Kumar et al 2008], games/animation [Xu et al 2008, Bayless & Stout 2006], and music [Beck et al 2011]. Extensions into the K-12 curriculum are beginning to occur, but the primary emphasis is on computing principles applied to a sampling of disciplines rather than immersion in an interdisciplinary domain.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SIGCSE'16, March 02–05, 2016, Memphis, TN, USA

Copyright © 2016 ACM. ISBN 978-1-4503-3685-7/16/03...\$15.00.

<http://dx.doi.org/10.1145/2839509.2844611>

Based on successful outcomes at the university level [Greenberg et al 2012, Greenberg et al 2013], the project described here explores high school level immersive experiences in *creative computation* -- a highly interdisciplinary area combining theory and methodology from Computer Science and Engineering with aesthetic principles and creative practices from the fine arts [Maida 2001].

Using a qualitative research approach we studied how creative computation can be applied at the high school level with an eye toward curriculum appropriate for both the existing CS AP A exam and the emerging CS Principles program. Creative computation also has potential for technology, art and media instruction within the common core standards for grades 6-12 (e.g. middle to high school [Wolz et al 2011]).

Two very different high schools (public:James Martin High School, Dallas, TX and private:Sidwell Friends School, Washington, DC) were chosen as the initial participants both because of their school cultures and for the marked difference in teacher pedagogical styles. These differences suggest that creative computation can be adapted across a broad spectrum of school cultures by teachers with a range of organizational and accountability constraints and levels of expertise. More formal analysis of a larger set of schools is underway.

James Martin High School (MHS), located in the Arlington School District in Dallas, TX, is a large (> 3000 student) public high school (grades 9-12) with rigorous standardization at the state level. Sidwell Friends School (SFS), located in Washington, DC is an east-coast elite private PK-12 day school with a total enrollment of approximately 1000 and a deep Quaker tradition.

Both teachers emphasize extensive student-initiated problem solving. Aaron Cadle (MHS) provides a highly structured, more traditional delivery of instruction style using a huge corpus of materials he developed; Darby Thompson (SFS) approaches the material through quick face-to-face delivery followed by extensive one-on-one and small group interaction. Both are experienced computer science teachers and hold undergraduate or higher degrees in computer science.

The two teachers adapted the college-level creative computation, CS1 course to a Level 3B course (proposed CSTA K-12 Curriculum [Seehorn et al 2011]) integrated into their specific curricula. As expected, they took different approaches based on their school culture. Both schools are (1) leaders in their

respective geographical areas, (2) have teachers who are proficient in Computer Science, and most importantly (3) have serious commitments from their school heads. We can claim now to have a corpus of exemplars of a Level 3B course that we believe offers an easy on-ramp for high school teachers with experience in computer science, information technology or media studies.

2. CREATIVE COMPUTATION FOR CS1

Creative Computation is an emerging discipline that requires a thorough grounding in both media arts and computing. Put simply, it is the discipline and theory of visual arts rendered through computer code. On top of programming principles students learn essential graphics operations such as transformations, iteration and randomization; algorithmic drawing, animations, basic physics, and interactivity. Areas of focus include creative use of image processing and emergent systems: two and higher-dimensional arrays are natural data structure choices for pixel-based image manipulations as well as implementing Cellular Automata such as Game of Life. Finally, text and data visualization round out the fusion of artistic design and computing.

Students engage in artistic creative acts to practice and master essential concepts. As a vehicle for teaching introductory computer science our design is consistent with a traditional Java-based CS1, however application of these core concepts is taught through highly individualized artistic expression. In other words, we teach the same material as the standard CS1 course, but instead of solving for roots of polynomials, simulating gas station/cash registers, or implementing graphics rendering techniques, we create art, interactive media and visualizations using algorithms and code.

The college level course upon which the high school experience is based comprises the following topics, in typical order of presentation during the semester. Topics 10-12 add advanced material appropriate for visualization and medium-to-large datasets:

1. Introduction to programming and drawing primitives.
2. Variables, data types and expressions
3. Control structures and simple physics simulations
4. Animation, Iterations
5. Using APIs and writing Functions
6. Coordinate systems, trigonometry, geometry and matrix transformations



Figure 1: Student submissions for assignment 1, after 1 week of lecture

7. Object-oriented Programming and Design
8. Arrays
9. Inheritance, abstract classes and interfaces
10. Recursion
11. Image processing and emergent systems (2D arrays)
12. Text and Data visualization
13. Transition to CS2 with full Java

Exercises designed for the course include a sequence of programming projects that utilize the concepts being introduced in the lectures as detailed above. Every exercise has a well-defined basic part and offers students the opportunities to explore open-ended creative projects. This is critical for the retention of student interest and motivation. Students take ownership of their work. Also, as a consequence, each submitted work is a unique program. No two programs are the same.

3. PROCESSING AND FISH TANKS

The curriculum assumes use of the Processing IDE (processing.org). It also assumes an art studio approach of iterative project development based on formal and informal critique. Students develop a visual portfolio of their work as they proceed through the course. Portfolio-based assessments enhance student awareness of the process of gaining and applying programming knowledge and skills especially during the Pilot III efforts being carried out during the development of the CS Principles course [Arpaci-Dusseau et al 2013].

Our perspective follows in the footsteps of John Maeda, who is both a formally trained artist and computer scientist. Inventor of the programming environment Design by Numbers, a precursor to Processing, he pioneered an approach to “Creative coding” that radically *recontextualized* computer code—from an applied math notation to a creative medium, on par with charcoal, paint, clay, etc. Creative coding is an exploratory and aesthetically driven approach, where students build visual designs and artworks iteratively as they expand their programs [Maida 1999].

Processing is built on top of Java with a simplified syntax and an API that focuses on graphics/media programming. It is a robust and full-featured language that has use both in the classroom and beyond; it is used widely in industry and growing quickly in popularity. It supports code writing in Java, Python and JavaScript. A Processing project is developed in a simple IDE that dispenses with the complexity of creating a complex



Figure 2: Class aquarium populated by students, spring 2012 collection of classes, instead using Java inner classes that provide

a gentle introduction [Koffman & Wolz 1999] to object-oriented programming.

For example, a student-favorite among the assignments is a collaborative class aquarium (virtual fish tank) project where students use OOP principles to design an animated (possibly interactive) sea creature of their choosing. Instructors provide an interface - the fish tank and generic fish as a super class or abstract class; and the students contribute customized implementations (sea creature class, physics class, etc). In this exercise they learn OOP principles: classes, objects, inheritance, polymorphism, and also team programming. The aquarium is then populated with student's creatures and displayed in a public space. Figure 2 provides an example, but doesn't do justice to the animated effect of the objects moving in its own way.

4. FROM COLLEGE TO HIGH SCHOOL

The curriculum outlined above has been in place within undergraduate introductory courses at Bryn Mawr College, Bryn Mawr, PA and Southern Methodist University, Dallas, TX for over six years. These institutions saw a significant increase in introductory computing enrollments prior to the more recent national surge. These surges are directly attributable to the creative computation approach (as reported in [Greenberg et al 2012, Greenberg et al 2013]). In addition, we report evidences of this context attracting women and non-STEM or undecided majors to computer science. 20% of Bryn Mawr's all-female freshmen class now takes CS1 (lottery still required), while SMU reports 41% female in Processing-based CS1 classes and 50% women majors in its new Creative Computation program.

Bryn Mawr College is a small, all-women's liberal arts college in the suburbs of Philadelphia that enrolls 1,300 undergraduates. The Computer Science Department offers a B.A. in Computer Science as well as a minor in Computer Science and in Computational Methods.

Southern Methodist University is a private university of 11,000 students near the vibrant heart of Dallas, TX. The university offers nationally competitive undergraduate, graduate and professional programs. The Creative Computation Program is supported through both the School of the Arts and the Department of Computer Science and Engineering housed in the School of Engineering and offers degrees in both Computer Science and Computer Engineering.

In both institutions Creative Computation has been successfully adopted to fit the local intellectual culture. Ten different instructors successfully offered our curriculum more than twenty times. As a curriculum intended to support the bridge from high school to college, a natural consequence was to investigate how it could be adapted to the cultures of high school. The remainder of this paper reports on the experience of bringing Creative Computation into high schools at opposite ends of the of a spectrum. Please visit the project website at www.cs.brynmawr.edu/visual for more information.

5. COACHING FOCUS AT SIDWELL FRIENDS SCHOOL

Sidwell Friends School (SFS) provides the kind of intimate, student-centered setting that is ideal for highly individual, rigorous instruction. The implementation of Creative Computation as elective subject matter at SFS provides a model for schools that can support a highly collaborative environment with close contact

between students, faculty and parents in which rigorous engagement by students is expected.

5.1 School Description

Sidwell Friends School is a private coeducational PK-12 Quaker school in Washington, DC. The school offers a rich and rigorous interdisciplinary curriculum designed to stimulate creative inquiry, intellectual achievement and independent thinking. Virtually 100% of students enroll in four-year colleges upon graduation. The students routinely attend a wide variety of colleges and universities including the full range of the most selective institutions in the country. At the start of this project in 2013, The Upper School (grades 9-12) offered computer science as an elective to 480 and had 62 students enrolled (16% female) in academic year 2012-2013.

The computer science curriculum at SFS offers a variety of semester-long courses ranging from introductory programming to data structures and algorithms, artificial intelligence and game development, to dynamic web design. The intention of this curriculum is two-fold: firstly, to provide an accelerated track for students who are already highly motivated to learn computer science and secondly to engage students who might never have considered computer science as a potential major after graduation.

5.2 Pedagogical Style and Curriculum

Darby Thompson has been teaching computer science at SFS for five years and developed the current curriculum from scratch. Previously, she spent five years teaching introductory CS courses at the George Washington University while pursuing her PhD in Computer Science focusing on robotics and user interfaces.

Thompson's teaching style and her school culture de-emphasize worksheet/small task exercises and emphasizes project work. This approach is a direct outcome of her experience with the 'Mindstorms' pedagogy of Seymour Papert. Her resources are organized into 'Labs.' The labs are available on request from the author, and the sequence from the 2014-2015 curriculum implementation is shown in Table 1.

Topics	Lab
1 Programming	001_Bouncing_Ball 01_Kindergarten_Picture
2 Functions, variables, basic loops, 2D Arrays	02_Image_Processing
3 Fundamentals, Control structures	03_Andy_Warhol 04_Green_Screen
4 Algorithm development	05_Finding_The_Robots_Ball
5 OOP:Classes, Polymorphism, animation	01_BouncingBall 02_SeaCreature
6 OOP and Design	03_SpaceInvaders
7 Abstraction Strings	04_DataVisualization
8 OOP:Interfaces	05_SwimmableObject 06_Paint
9 OOP:Inheritance	07_SeaCreatureInheritance
10 Recursion	08_HanoiTower
11 Algorithm development, OOP: Encapsulation	09_RobotMaze

Table 1: Articulation of Curricular with High School Labs

Topics 1-4 in Table 1 are covered in an introductory semester-long course, immediately following a 4-week introduction to programming in Python. Topics 5-11 are covered in the follow up

intermediate semester-long course with the emphasis on learning OOP. Each course has an open-ended final lab. Her intermediate course final lab is a side scroller/platform game intended to pull together concepts from Object Oriented Programming.

Thompson presents material in a lightening-lecture fashion as the start of a new project. Students are then encouraged to collaborate to develop the basic concepts and skills to complete the project with direct one-on-one support from Thompson. Assessment occurs through rubric-based project evaluation. Student work is displayed throughout the school on large-screen monitors.

5.3 Outcomes Academic Year 2013 – 2014

Sidwell Friends School offers an Introduction to CS course. In the fall semester (2013) this course continued to use Python to explore variables, conditionals, loops and functions. Creative Computation was incorporated into a second semester course: Intermediate Computer Science (Spring 2014). Students engaged in the projects with an emphasis on object-oriented programming and the Java Language. Enrollment in Intermediate CS was 33.

Creative Computation also had a place in Advanced CS. Students used Processing to create data structures (Linked Lists, Trees) and built in data structures to program standard sorting, searching and graphing algorithms. Of the 33 enrolled in Intermediate CS, 8 were graduating seniors, and 22 of the remaining 25 students continued onto Advanced CS giving a retention rate of 88%.

Artificial Intelligence and Game Design is an advanced course, which in this academic year was populated by students who did not have a Creative Computation background. However students intermittently used Processing to develop AI behaviors and program AI players for games such as 4x4 Tic-Tac-Toe.

Ratio of female students doubled to 32% from 16% in 2012-2013.

5.4 Outcomes Academic Year 2014 – 2015

Creative Computation migrated into the Introduction to CS Course and contributed to a dramatic increase in interest in computer science. The curriculum was modified to transition from Python to Processing after the first 4 weeks of the course instead of at the end with positive feedback from the students. In Fall 2014, Thompson taught 4 sections of Introduction to CS with an enrollment of 61 students and 1 section of Advanced CS with a full enrollment of 18 students. The courses were over enrolled. An extra section of Introduction to CS was added. Students faced a lottery to determine if they could take the class. In the spring, 48 students continued on from the introductory class to take Intermediate CS. The 78% does not reflect a significant downturn in retention, but rather was impacted by the need to limit the number of sections offered.

Ratio of female students again improved to 37% in 2014-2015. Most encouragingly, current female ratio reached 52% in 2015-2016. Fall 2015 will witness our first section with more female students than males and the preliminary signups for Advanced CS for Fall 2015 are 50/50 for the first time since the creation of the program in 2010.

5.5 Curricular Change – Lessons Learned

Creative Computation has been fully embraced at Sidwell Friends School. Initial concern with including Creative Computation in the introductory course was rapidly mitigated as students in the intermediate course produced stunning projects that excited members of the school-wide community. The doubling of interest in computer science between the two years is, of course,

significant, as well as the rapidly increasing percentage of female students. It should be noted again that Thompson built the computer science program over a five-year period. Creative Computation sparked the interests of students school-wide.

Students at SFS are noticeably more vocal and confident about their work in the CS courses since the transition to Creative Computation. The approach is more natural to apply to group projects than traditional text-based assignments. In addition, regular reflection is required of the students. Their projects are demonstrated to their classmates and teacher in class, but also to the entire school via a display in the hallway. This has increased the visibility of the CS program at SFS. Although SFS takes a project-based approach, repetition and drills of the basic concepts still play an important role in the course. The open-endedness and creative aspect of the assignments have allowed the courses to be successful for students with a very wide range of abilities.

6. MATERIAL SUPPORT LEARNING AT JAMES MARTIN HIGH SCHOOL

James Martin High School (MHS) has a very different school culture than Sidwell Friends School. It is a large public high school that accommodates a diverse population with a full range of socio-economic needs. As such, expectations for achievement require significant reinforcement via extensive use of highly structured materials. The computer science program at MHS is targeted toward Advanced Placement (a metric used to evaluate the quality of the education unnecessary at schools like SFS). The school culture, in combination with the personal style of CS teacher Aaron Cadle, result in an equally successful experience, but one with significantly more support materials.

6.1 School Description

James Martin High School enrolls over 3300 students in grades 9-12 and is the largest public secondary school in the Arlington independent school district in Dallas, TX. 68% of students who graduate attend a four-year college. In order to serve such a large and diverse population, the school offers over 20 different Advance Placement classes. At the start of this project in 2013, 140 students were enrolled in the Computer Science program (28% female). The CS program consists of three courses at MHS: Pre-AP Computer Science (CS0 at college level), AP Computer Science (CS1 at the college level) and Advanced Computer Science III. Due to early introduction of Processing into the curriculum independent of this project starting in 2010, MHS has enjoyed a higher percentage of female students in the CS program, especially in Pre-AP Computer Science, where much of the Processing instruction concentrated before the start of this project. For the three years 2010-2013, Pre-AP Computer Science consistently enrolled 32%-34% female students, but upper level CS courses rarely saw any female participation.

6.2 Pedagogical Style

Aaron Cadle holds a B.S. in Computer Science from Texas A&M University. He is currently in his 7th year teaching computer science at MHS. Beginning with 45 students and a very traditional CS curriculum in 2008-2009, MFS has grown the program to 140 students as an early adopter of Creative Computation. Beginning in 2010 he redesigned the introductory course (Pre-AP Computer Science) to include elements of Creative Computation using Processing, game development using Scratch, object oriented programming using Jeroo, and hardware design using Logg-O. Currently his Pre-AP Computer Science

course covers the first three topics in Table 1. Students typically take this course in 9th grade.

This combination allows students unfamiliar or uncomfortable with computer science to obtain a general understanding while expecting enough rigor to prepare students for a more advanced second year course (AP Computer Science). All second year students are required to take the AP CS exam. AP students boast a 94% passing rate (3 or better) with 80% of those students making a 4 or better on the exam. A third year course (CS III) explores data structures for one semester while focusing on creative computation with Processing and Microsoft Kinect during the second semester.

Cadle employs teaching strategies that support large-class, large-school, diverse-population techniques. He has developed a corpus of approximately 100 individual items in order to keep students engaged and accountable. These include:

- Slide decks: for full class problem solving; both outlines for live writing and completed solutions for review.
- Student worksheets and quizzes, tests.
- Warm ups: quick puzzles assigned to students to solve at the start of class.

Following highly structured presentations and individual assignments via worksheet, he provides ‘labs’ that are similar in focus and style to Thompson, and indeed provide one-on-one coaching support from the instructor. Assessment occurs through regular quizzes and tests. Materials are available on request from the author.

6.3 Outcomes Academic Year 2013 – 2014

Students used Processing exclusively to implement Creative Computing projects in Pre-AP Computer Science. The syllabus covered basic programming concepts in including variables, conditionals, loops, but also advanced into CS1 territory with introductory object-oriented concepts of methods, and classes. Enrollment for Pre-AP Computer Science reached 125 students (29% female) for the 2013-2014 school year.

In AP Computer Science, Processing was used intermittently to reinforce topics traditionally covered in a CS1 course (Arrays, 2D Arrays, Objected-oriented programming, ArrayLists, Inheritance). 60 students (15% female) took AP Computer Science during the 2013-2014 school year.

Advanced Computer Science III – Students used Processing and a Microsoft Kinect to create motion control games. Students spent the first semester learning the fundamental data structures for Computer Science (LinkedLists, BinaryTrees, Sets, Maps). The second semester students learned how to interact with a Microsoft Kinect through Processing. They spent the entire second semester developing an interactive game using the motion capture features of the Kinect and the graphical abilities of Processing. Enrollment for Advanced Computer Science III was 7 students (14% female) for the 2013-2014 school year.

The number of students taking Computer Science grew from 140 students for the 2012-2013 school year to 193 students for the 2013-2014 school year due to the full-scale introduction of Creative Computation in the classroom. AP CS also more than doubled to 60 from 27 in 2012-2013. A second CS teacher was hired to help with the course load.

6.4 Outcomes Academic Year 2014 – 2015

183 students took computer science in 2014-2015, despite staffing constraints that forced enrollment limits for the first time in Pre-AP Computer Science. 2014-2015 was a building year for MHS’s CS program while bringing the second CS teacher to speed. Courses already implemented using Creative Computation and Processing only instituted incremental changes to their curriculum. The second CS teacher taught all four sections of Pre-AP Computer Science and readily adopted Processing as her teaching tool of choice. Enrollment for Pre-AP Computer Science was 107 students (32% female) for 2014-2015, down slightly from 2013-2014 entirely because of the enrollment caps. AP Computer Science enrolled 47 students (13% female), and enrollment for Computer Science III increased from 7 students in 2013-2014 to 29 students (7% female) for 2014-2015.

In addition, currently AP CS in 2015-2016 enrolls 29% female students (45 total). It is the first time in history that AP CS has approached the critical threshold of 30% women.

6.5 Curricular Change – Lessons Learned

Cadle had a head-start on adapting Creative Computation to high school curricula, using Processing in the classroom as early as 2010. However, in the last two years the focused emphasis on the Processing IDE and artistic design significantly influenced the increased interest in and sustainability through computing course offerings. Note that MHS courses are year-long. Sustained enrollment from the Pre-AP to AP course was impacted by limits on section offerings and class size. However the significant outcome is the tripling of the number of students enrolled in Computer Science III. Of the 60 students in AP Computer Science from 2013-2014, 27 chose to continue to CS III in 2014-2015. A 45% retention rate at an academically rigorous public high school is extraordinary, given the competing demands on student attention and time. The increasing participation of female students in upper level CS courses beyond Pre-AP is also very encouraging.

Processing is an event-driven language that requires a “re-thinking” of traditional labs. The first program is no longer “Hello World” but a graphical picture – possibly a smiley face, a robot, or a skyline? Processing allows teachers to contextualize their labs to their environments. Processing allows for more individual exploration as well due to the simplicity of starting a program. Minimal syntax is needed to make a sketch work and the examples on the processing.org website are easy to understand. Some concepts, like recursion, are more readily engaged with students willing to work on a recursive graphical lab much longer than a traditional recursive lab (e.g. code the Fibonacci sequence). Typical worksheets should be redesigned as well. A worksheet on loops now requires students to draw multiple shapes rather than just a sequence of numbers. A combination of both old problems and new Processing problems is a great way to prepare students for a subsequent course that may be more traditional.

7. ANALYSIS

James Martin High School and Sidwell Friends School teachers have successfully adapted the college-level creative computation model to the following course:

- An introduction to computer science course that starts with 4 weeks of Python following by Processing. The course covers materials traditionally associated with CS1 with an introduction to classes but without inheritance.

- An introduction to computer science course that uses Processing exclusively and covers material traditionally associated with CS0.
- An advanced computer science course that parallel expectations for a traditional CS1/CS2 experience exclusively taught in Processing.
- An AP CS course that requires students to sit for the AP exam. Processing and Creative Computation projects are used intermittently because this approach is not well aligned with the test.
- An Artificial Intelligence and Game Design course that has evolved into a significant example of Creative Computation that goes beyond the fine arts.
- An advanced computer science class that covers material traditional found in CS2 as well as providing opportunities for kinesthetic game design.

These courses are taught in two schools at the extreme end of the spectrum of high school cultures, using pedagogy that is similar in the emphasis on project work, but markedly different in use of materials and presentation style.

In both cases enrollment in Computer Science increased significantly, as did female participation. Because the research to date is qualitative on a tiny sample size we cannot draw conclusions about either causality or coincidence. Nation-wide, the CS Principles movement is providing evidence that contextualized computing in safe, uncomplicated IDEs supports increased participation in computer science classes. The conclusion we can draw from this experience is that Creative Computation has a place in the arsenal of appropriate curricula.

8. NEXT STEPS

The authors are concluding a National Science Foundation project in which over a dozen high school teachers have been introduced to our high school Creative Computation curricula. Four ‘mini grants’ (two public and two private schools) have been awarded to high school teachers to adopt the course and adapted curricula to their own institutions. We anticipate providing more substantive evidence of adaptability to high school within the coming year.

The authors are also developing a publicly available indexed repository that will allow instructors to locate relevant materials by media as well as computing topic, traditional course relevance, and teaching style. Options will be available for curated contributions to the repository.

Finally we address the question of the relevance of this curriculum to (1) current advanced placement preparation, (2) as general preparation for undergraduate level computer science. Creative Computation was used in classes that span 9th – 12th grade, exposing students to material in the CS AP (and in one instance directly preparing them for the exam.) Our experience was not merely ‘pushing down’ a college level course, but in adapting curricula to the expectations of high school curricula. Our next step is to understand the role these curricula might play in offering the new ‘CS Principles AP’ as more information emerges.

Regarding the question of transition to an undergraduate experience: the Computer Science Teachers Association (CSTA.org) takes a strong stand on the need to bring computing into the K-12 curriculum. One size does not fit all. The experiences related here suggest a need for rigorous study of impact K-12 curricula for developmentally appropriate experiences. Our report suggests that Creative Computation has a definite place in the high school experience.

9. ACKNOWLEDGMENTS

This project is supported in part by grants from The National Science Foundation (DUE-0942626, DUE-0942628, DUE-1323463, DUE-1323305, CCF-0939370, and CCF-1140489), and from Bryn Mawr College and Southern Methodist University.

10. REFERENCES

- [1] [Arpaci-Dusseau *et al* 2013] A. Arpaci-Dusseau, M. Bauer, B. Franke, J. Griffin, R. Morelli, O. Astrachan, M. Carrell, C. Gardner, R. Kick, D. Muralidhar, D. Barnett, R. Dovi, J. Gray, A. Kuemmel, and R. Brook Osborne. Computer Science Principles: Analysis of a Proposed Advanced Placement Course. In *Proceedings of SIGCSE 2013*. ACM Press 2013.
- [2] [Bayless & Stout 2006] Jessica D. Bayless and Sean Strout. Games as a “Flavor” of CS1. In *proceedings of SIGCSE 2006*. ACM Press 2006.
- [3] [Beck *et al* 2011] Robert E. Beck, Jennifer Burg, Jesse M. Heines, and Bill Manaris. Computing and Music: A Spectrum of Sound. *Special Session, SIGCSE 2011*. Dallas, TX, March 2011.
- [4] [Cassel & Wolz 2013] Cassel, L. and Wolz, U. Interdisciplinary Computing, Successes and Challenges. In *Proceedings of SIGCSE 2013*. ACM Press 2013.
- [5] [Greenberg *et al* 2012] Ira Greenberg, Deepak Kumar and Dianna Xu. Creative Coding and Visual Portfolio for CS1. In *Proceedings of SIGCSE 2012*. ACM Press 2012.
- [6] [Greenberg *et al* 2013] Ira Greenberg, Dianna Xu, and Deepak Kumar. *Creative Coding and Generative Art in Processing 2.0*. friends Of ed/Apress 2013.
- [7] [Guzdial 2004] Mark Guzdial. *Introduction to computing and programming with Python: A Multimedia Approach*. Prentice-Hall, 2004.
- [8] [Koffman & Wolz 1999] Koffman, E. and Wolz, U. CS1 using Java language features gently. In *Proceedings of SIGCSE/ITiCSE 1999*. ACM Press 1999.
- [9] [Kumar *et al* 2008] Deepak Kumar, Doug Blank, Tucker Balch, Keith O’Hara, Mark Guzdial, Stewart Tansley. Engaging Computing Students with AI and Robotics. *Symposium on Using AI to Motivate Greater Participation in Computer Science*, 2008.
- [10] [Maida 2001] John Maeda *Design by Numbers*, MIT Press 2001.
- [11] [Seehorn *et al* 2011] K-12 Computer Science Standards, CSTA 2011.
- [12] [Summet *et al* 2009] Jay Summet, Deepak Kumar, Keith O’Hara, Daniel Walker, Lijun Ni, Doug Blank, and Tucker Balch. Personalizing CS1 with Robots. In *Proceedings of ACM SIGCSE 2009*. ACM Press 2009.
- [13] [Wolz *et al* 2011] Wolz, U., K. Pearson, S.M. Pulimood, M. Stone, M. Switzer. Computational Thinking and Expository Writing in the Middle School: A novel approach to broadening participation in computing, *Transactions on Computing Education*, 2011, Volume 2, article 9.
- [14] [Xu *et al* 2008] Dianna Xu, Douglas Blank, and Deepak Kumar. Games, Robots and Robot Games: Complementary Contexts for Introductory Computing Education. In *Proceedings of Third International Conference on Game Development in Computer Science Education (GDCSE’08)*, 2008.