

EVALUATING CREATIVE CHOICE IN K-12 COMPUTER SCIENCE
CURRICULUM

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science

by
Kirsten Mork
June 2019

© 2019
Kirsten Mork
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Evaluating Creative Choice in K-12 Computer Science Curriculum

AUTHOR: Kirsten Mork

DATE SUBMITTED: June 2019

COMMITTEE CHAIR: Zoë Wood, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Theresa Anne Migler-VonDollen, Ph.D.
Assistant Professor of Computer Science

COMMITTEE MEMBER: Aaron Keen, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: John Wilcox
Fifth Grade Teacher at Peabody Charter Elementary School

ABSTRACT

Evaluating Creative Choice in K-12 Computer Science Curriculum

Kirsten Mork

Computer Science is an increasingly important topic in K-12 education. Ever since the “computing crisis” of the early 2000s, where enrollment in CS dropped by over half in a five year span [19, 74], increasing research has gone into improving and broadening enrollment in CS courses. Research shows the importance of introducing CS at a young age and the need for more exposure for younger children and young adults alike in order to work towards equity in the field [43, 69]. While there are many reasons for disinterest in CS courses [19], studies found one reason young adults do not want to study CS is a perception of it being tedious and lacking opportunities for creativity [74]. Making more creative assignments is one way to try and engage more students who may not feel like stereotypical computer scientists.

This thesis focuses in on the effects of creative choice in CS curriculum on students’ self-efficacy, engagement/preferences, and performance. It aims to capture the effects of creative choice on a range of K-12 students of varying demographics in order to make CS more accessible for everyone. The first half of the thesis experimentally validates the effects of creative choice in existing 5th grade CS classes. We created two variants of worksheets for the students - creative worksheets and rigid worksheets. After distributing these worksheets, surveys, and quizzes, we found students still feel a sense of ownership with limited versions of creative choice and benefit from a blend of creative and rigid instructions. In addition, student performance was not affected by our different treatments. The second half of the thesis adapted and launched the fifth grade curriculum to a new demographic, teaching the course to Juvenile Hall students. Student surveys and reports from their teacher showed this class had a positive impact and was well received by students and staff. We found students would

prefer a longer class next iteration, as this one only extended five weeks. Future work will be needed to experimentally evaluate the specific impact of creative choice in this new demographic.

ACKNOWLEDGMENTS

Thanks to:

- Zoë Wood, for being my wonderful advisor and for all of her hard work and support
- Theresa Migler-VonDollen, Aaron Keen, and John Wilcox for being on my committee and helping so much along the way
- John Wilcox and Peabody Charter Elementary School, for letting me take so much lab time for this thesis
- Robyn Morton and all the staff at Restorative Partners, for being supportive of our class and getting my volunteers cleared so quickly
- Greg Murphy, for being such an amazing teacher and for letting us come into his classroom to teach our course
- All of the staff at Juvenile Hall, for being so supportive and helpful
- Timothy Wong and Erik Mork, for volunteering at Juvenile Hall with me. I could not have taught the course without you both. (Also thanks to Erik for being the best brother)
- Timothy Wong, Amy Lewis, and Kole Yager, for being Zoë's kids with me
- My parents, for listening, supporting and always being there
- Bobby Boyd, for listening to me brainstorm for my thesis all year
- Anne and Paige Anaya, for being amazing elementary school teachers in my family and for all of the ideas
- Andrew Guenther, for uploading this template

TABLE OF CONTENTS

	Page
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER	
1 INTRODUCTION	1
1.1 Problem	4
1.2 Context	5
1.3 Contribution	6
2 BACKGROUND	9
2.1 Self-Efficacy	9
2.1.1 Defining Self-Efficacy	10
2.1.2 Importance of Self-Efficacy	10
2.1.3 Testing for Self-Efficacy	11
2.2 Creativity	12
2.2.1 Creativity in CS Education	12
2.3 Designing a Curriculum to Engage Middle and High School Students	13
2.3.1 Curricular Approaches	14
3 RELATED WORK	16
3.1 Code.org	16
3.2 Bootstrap	17
3.3 MyCS	19
3.4 Peabody Charter Outreach	20
3.5 Processing Classes	22

3.6	Introductory Computer Science Courses with Restorative Partners . .	22
4	FIFTH GRADE CURRICULUM	24
4.1	Intervention One	26
4.1.1	Design	27
4.1.2	Research Questions - Design, Results, and Threats to Validity	31
4.1.3	Summary	42
4.2	Intervention Two	43
4.2.1	Design	44
4.2.2	Research Questions - Design, Results, and Threats to Validity	53
4.2.3	Summary	64
5	JUVENILE HALL CURRICULUM	66
5.1	Design	66
5.1.1	What: Introductory Course	67
5.1.2	Who: Demographics	67
5.1.3	How: the Curriculum	68
5.2	Implementation	71
5.2.1	Setup	72
5.2.2	Lessons	72
5.3	Validation	79
5.3.1	Research Approval	79
5.3.2	Surveys	79
5.3.3	Quantitative Results	79
5.3.4	Qualitative Results	83
5.3.5	Recommendations	85
5.3.6	Threats to Validity	85

6	CONCLUSION AND FUTURE WORK	87
6.1	Fifth Grade Curriculum	87
6.1.1	Future Work	88
6.2	Juvenile Hall Curriculum	89
	BIBLIOGRAPHY	91
	APPENDICES	
A	IRB PROPOSAL	101
B	IRB MODIFICATION REQUEST	107
C	PEABODY-INTERVENTION 1: CC MOUSE	111
D	PEABODY-INTERVENTION 1: CC SNOWMAN	126
E	PEABODY-INTERVENTION 1: NO CC MOUSE	141
F	PEABODY-INTERVENTION 1: NO CC SNOWMAN	145
G	PEABODY-INTERVENTION 1: POST PRACTICE SURVEY	150
H	PEABODY-INTERVENTION 2: CC CONDITIONALS	153
I	PEABODY-INTERVENTION 2: NO CC CONDITIONALS	165
J	PEABODY-INTERVENTION 2: Surveys	176
K	JUVENILE HALL SYLLABUS	193
L	CVA TEACHER’S REPORT ON JUVENILE HALL CLASS	195
M	CVA Pre-Course Survey	197

LIST OF TABLES

Table	Page
1 Count of Student Preferences in No CC Treatment (Note: Students All Received the Same Treatment for the Variable Sheets).	63
2 Count of Student Preferences in CC Treatment (Note: Students All Received the Same Treatment for the Variable Sheets).	63
3 Number of Students (of Six) Preferring a Game-Design Course, Computational Art Course, or Either.	80
4 Pre-Survey Concerning Computer Science as a Career (7-Point Likert Scale)	81
5 Pre-Survey Concerning Desire to Learn Programming Topics (7-Point Likert Scale)	81
6 Pre-Survey Concerning Creativity (7-Point Likert Scale)	81
7 Averages of Student Responses from Selected Pre-Survey Questions (7-Point Likert Scale)	82
8 Averages of the Post-Survey Results for the Six Students Present on the Last Day of Class (7-Point Likert Scale)	82
9 Comparison of Pre and Post Surveys for the Students Who Took Both (Other Students Joined Late or Left Early) (7-Point Likert Scale)	83

LIST OF FIGURES

Figure		Page
1	Gender at 5 Institutions with Most Students Graduating in Software Engineering, 2016 [7]	2
2	Diversity and Gender of Awarded Bachelor's Degree in Software Engineering, 2016 [7]	2
3	Racial Diversity of Awarded Bachelor's Degree in Software Engineering, 2016 [7]	3
4	Peabody Charter Students' Self-Reported Enjoyment of Math, Enjoyment of Art, and Interest in Computer Programming (5-Point Likert Scale).	7
5	Teenage Perceptions of CS [74]	13
6	Online Processing Editor [9]	20
7	Sample Curriculum [71]	21
8	Student Enrollment at Peabody Charter by Ethnicity.	24
9	Student Enrollment at Peabody Charter by Other Demographics.	25
10	The First Instruction for Selecting the Background Color for the Creative Choice Worksheet.	27
11	The First Instruction for Making the Background Color for the Non-Creative Worksheet.	28
12	The Two Images Made by Each Student During <i>Intervention One</i> (Though One of the Images Varied Depending on Which One Was on the Creative Choice Worksheet for a Given Student).	29
13	Checked Reasons for Worksheet Preference	32
14	Free Response for Worksheet Preference	34
15	Free Response Worksheet Style Preferences (Following Instructions or Picking Options)	36

16	Average Count of Created Options by Worksheet Style Preference (Picking Options or Following Instructions).	41
17	<i>Conditionals Part One</i> : Example Picture Created by a Fifth Grade Student in the Creative Treatment Group.	46
18	<i>Conditionals Part One</i> : The Two Circle Puzzles the Students in the Non-Creative Treatment Were Asked to Make.	46
19	Conditionals Part Two Worksheet Segment: Example Challenge Assigned to Students in the Creative Choice Treatment.	47
20	Conditionals Part Two Worksheet Segment: Example challenge Assigned to Students in the Non-Creative Choice Treatment.	48
21	5-point Likert Scale Used for Efficacy Surveys.	49
22	Oneway Analysis of Variable Performance Scores by Lab Period. . .	53
23	Analysis of Variance on Variable Performance Scores by Lab Period.	53
24	Oneway Analysis of Variable and Conditional Performance Scores by Self-Regulatory Efficacy rankings of High and Low.	56
25	Oneway Analysis of Variable and Conditional Performance Scores by Variable and Conditional Self-Efficacy Scores.	57
26	Oneway Analysis of Conditional Performance Scores by Lab Period (All Periods - Monday-1:55 and Thursday-8:30 Labs Were Given the Creative Treatment)	59
27	Analysis of Variance on Variable Conditional Scores by Lab Period (All Periods - Monday-1:55 and Thursday-8:30 Labs Were Given the Creative Treatment)	60
28	Analysis of Variance on Variable Conditional Scores by Treatment (Creative and Non-Creative) (P-Value of 0.1783 Shows the Mean Difference is Not Statistically Significance)	60
29	Oneway Analysis of Conditional Performance Scores by Thursday Lab Period (8:30 Lab Receiving Creative Treatment and 9:50 Receiving Non-Creative).	61

30	Oneway Analysis of Conditional Self-Efficacy Scores by Treatments (Creative and Non-Creative) for All Classes (Left Image) and for Just the Thursday Classes (Right Image). (The P-Values of 0.6970 (Left) and 0.8206 (Right) Indicate No Significant Difference in Efficacy by Treatment)	61
31	First Assignment for Students to Start Planning Their Game. . . .	73
32	Backgrounds Students were Given to Chose from While Practicing Function Calls.	75

Chapter 1

INTRODUCTION

It is a commonly held belief that there is need for greater diversity and acceptance in the tech industry [20, 22]. However, while the belief is common across academia, solutions are less fleshed out. Questions arise, such as how and where to make changes? Some might look to industry as a starting place; however, though industry is where a lot of the discrepancies lie, by the time individuals are being hired into industry positions, the pool of qualified workers is not necessarily a diverse one. The next logical step is to examine college enrollment and retention within tech-related degrees. As could be expected, there are huge demographic gaps in students graduating with Bachelor's Degrees in technical majors. In 2016, for example, white males made up a majority of Software Engineering graduates, as seen in Figures 1, 2, and 3 [7]. Women are shown to make up lower percentages of computer majors than other STEM majors such as biology [23]. Underserved, intersecting groups face numerous challenges that more privileged students never begin to face [20, 51, 57]. Colleges have many issues concerning inclusivity, indicating their responsibility in making CS more equitable.

There are numerous steps colleges should take in increasing diversity and retention within CS. After a huge dip in computer science enrollment in the early 2000s, dubbed the “computing crisis” by some, more research went into increasing both enrollment as a whole and diversity within computer science degrees [19, 74]. Some research, for example, has found success in “designing for diversity”, specifically within the introductory courses [24]. While these steps have, in fact, created a more inclusive environment, there is now even more need to continue to create accessible curriculum

Gender Imbalance for Common Institutions for Computer Software Engineering Majors

The closest comparable data for the 6 Digit Course Computer Software Engineering is from the 2 Digit Course Engineering.

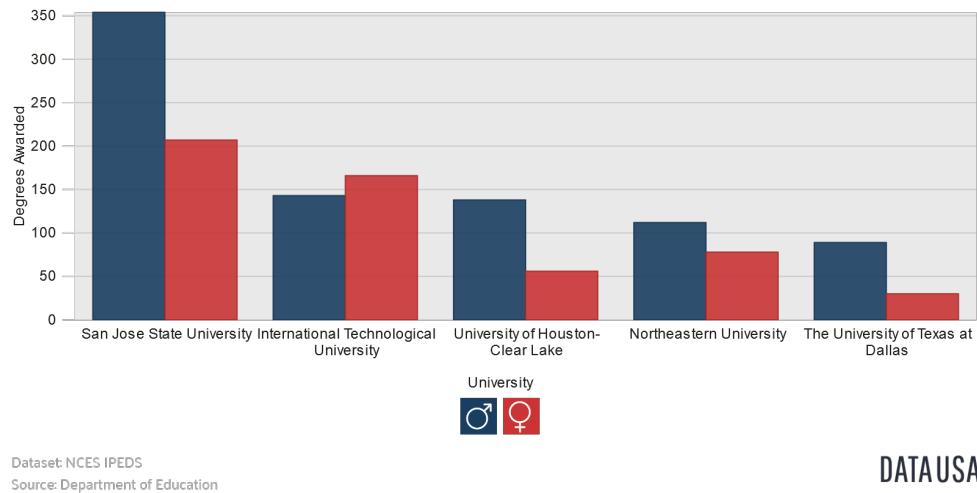


Figure 1: Gender at 5 Institutions with Most Students Graduating in Software Engineering, 2016 [7]

Race & Ethnicity by Gender for Computer Software Engineering Majors

The closest comparable data for the 6 Digit Course Computer Software Engineering is from the 2 Digit Course Engineering.

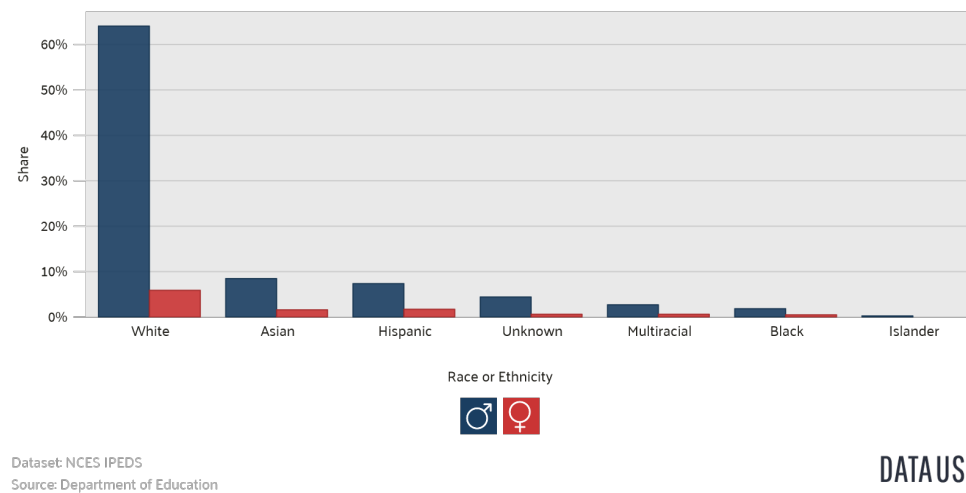
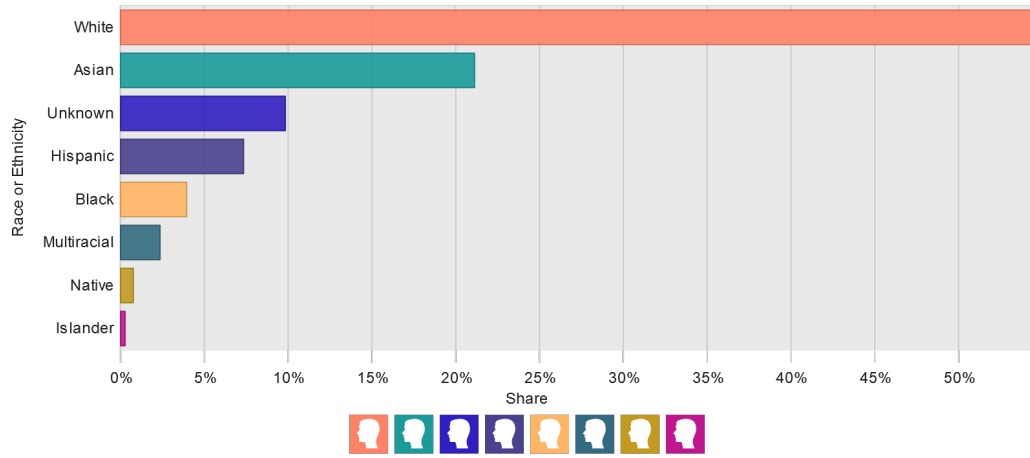


Figure 2: Diversity and Gender of Awarded Bachelor's Degree in Software Engineering, 2016 [7]

Race & Ethnicity by Degrees Awarded for Computer Software Engineering Majors

The closest comparable data for the 6 Digit Course Computer Software Engineering is from the 2 Digit Course Engineering.



Dataset: NCES IPEDS
Source: Department of Education

DATAUSA:

Figure 3: Racial Diversity of Awarded Bachelor's Degree in Software Engineering, 2016 [7]

for these students and others coming in [29]. However, while college-level change is important, it is neither sufficient nor most important in creating equity within software engineering.

While retention is an important focus for colleges and universities, colleges and universities cannot solve the problem of *increasing* diversity in isolation - they can merely attempt to maintain what they have. If there is not a diverse group of students applying for technical majors, how can there possibly be diversity within the major? As of 2018, under 30 percent of the students who took the AP CS exam were female, and just over 20 percent of the students were underserved minorities. There are quantifiable performance differences between students taking introductory college CS classes with prior CS experience versus those without [70], yet certain groups of students are not enrolling in high school CS courses. The problem of inequity within college comes from inequity before college. Thus the solution is to make changes pre-college.

Pre-college, however, is vague - do high school students merely need to be exposed to CS, or should exposure come earlier? Research shows that students as young 13 and 14 already have predictive inclinations to which career they will have, implying the importance of exposure to computer science before then [37]; clearly, to reach students early enough to make a difference, action is needed in the entire K-12 space. In addition, computer science stereotypes are embedded during these younger ages as well. In regards to the gender gap, studies show it begins before college and is perpetuated by stereotypes of the field and in the classroom [23, 48]; one study found that, before implementing computer science curriculum into the school, most fourth grade students associated computer science with males working alone [36]. If students are not given equal opportunity to K-12 computer science curriculum, and if stereotypes are not fought at younger ages, these students are placed at a disadvantage.

1.1 Problem

As computer science is being integrated into K-12 curriculum, it is important to consider the following: creating equal access in distribution, building equity into the curriculum itself, and designing age appropriate material [25]. K-12 computer science education is of utmost importance in increasing equity in college and beyond. However, just as in the upper levels of computer science, there has been extensive evidence of racial, socioeconomic, and gender inequalities in K-12 CS education [6]. Since stereotypes start so young, and early exposure to computer science is an important factor in a child pursuing a computer science degree, one of the most important ways to broaden diversity at the higher levels is to not only teach CS in the K-12 populace, but to create equal access and actively fight stereotypes in this age range as well. Young students cannot just be taught in the same way they have been and

expect a difference - students must be taught in a way that fights stereotypes of the field [23].

In addition to reaching more students, we have to design curriculum in a way to meet each group of students' specific needs. If curriculum is taught to students for which it was not designed, it could be more harmful than helpful in encouraging a computer science career. One of the areas to specifically design for is age, as K-12 is a broad range of students. For example, research shows the difference of capabilities within 4th, 5th, and 6th grade computer science classes and why it is important to target different concepts at these different grade levels [37, 74]. In addition, young adults have their own reasons for liking or disliking computer science, and should be taught in a way to tailor to their preferences [32]. As summarized by the K-12 Computer Science Framework Steering Committee: "The learning accommodations and curricular modifications demonstrate that established techniques for differentiation instruction can be readily applied in computer science to engage all students" [25]. Research is needed to see what engages different groups so curriculum can be modified appropriately.

1.2 Context

The two groups of students we specifically worked with during our research were within the broader context of K-12 CS education. First, we worked with a fifth grade computer science lab where four classes (of 107 students total) each met once a week at Peabody Charter Elementary School in Santa Barbara. Second, we taught a five-week course for middle and high school students at San Luis Obispo County's Juvenile Hall. The fifth grade intervention was conducted within an existing curriculum, which has students working for an hour in the computer lab every week on

projects introducing coding via simple computational art exercises. The Juvenile Hall course was a completely new intervention based on the fifth grade curriculum, which introduced programming by building a simple 2D game. This curriculum included substantial modifications to fit the needs of this target population.

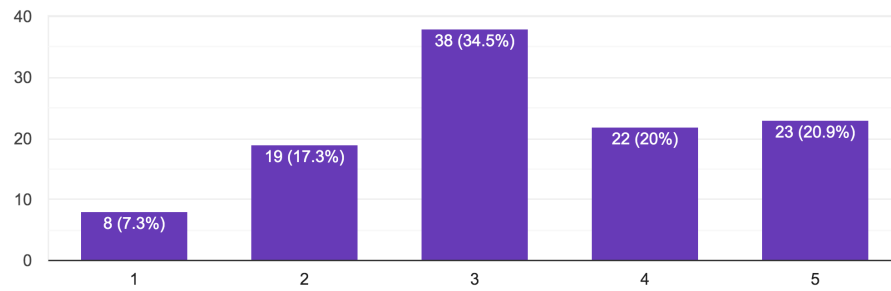
As mentioned above, it is important to not only teach these ranges of students, but also to create the most engaging curriculum for each of their needs. Many programs and organizations are already working in K-12 education, such as code.org and Bootstrap [3, 6]. There are many techniques being used to try to engage and teach young students better - groups have tried block-based coding, online modules, and more [28, 37, 62, 32, 64]. Within the context of K-12 curriculum, we specifically want to look at fostering creativity through course material, as this seems like a promising way to engage young students. One inspiration for this idea was a survey given to the fifth grade Peabody Charter students at the beginning of the school year. They were asked to rank how strongly they enjoyed learning math, enjoyed learning art, and were interested in computer programming. Students most strongly indicated an enjoyment of art (as seen in Figure 4). An inherent element of visual art is creativity and design. Since designing and choosing visual elements is a natural extension of making art, we wanted to utilize the creative choice found in art and game-first approaches to engage different groups of students. Other research validates that creativity is one way to engage different ages of students [64, 72, 74], strengthening our desire to research creative choice within our CS curriculum.

1.3 Contribution

The contribution of this thesis is improving, spreading, and designing curriculum with the goal of empowering students to pursue computer science through creativity.

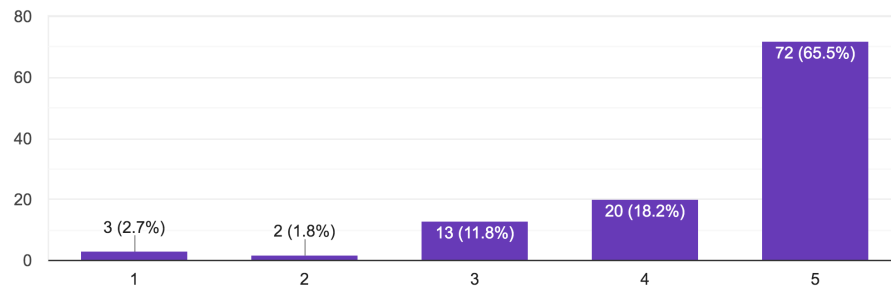
Do you enjoy learning math?

110 responses



Do you enjoy creating art?

110 responses



Do you think computer programming is interesting?

110 responses

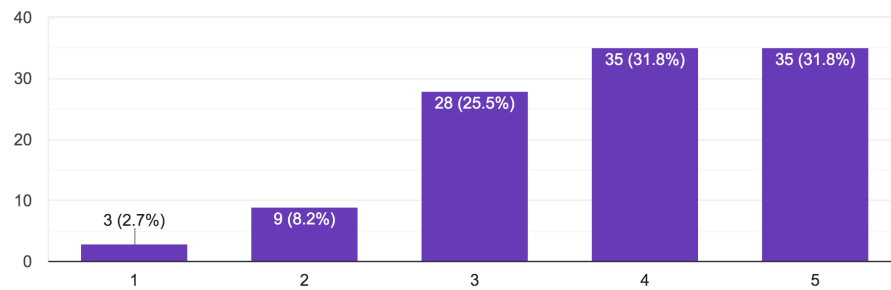


Figure 4: Peabody Charter Students' Self-Reported Enjoyment of Math, Enjoyment of Art, and Interest in Computer Programming (5-Point Likert Scale).

First, half of the thesis was conducted with the fifth-grade computer lab at Peabody Charter. Within this context, we isolated and tested how creativity and how creative choice benefits or hinders student learning within CS. We validated this aspect of the curriculum to try to make the best learning experience for students, keeping in mind their personal preferences, effects on self-efficacy, and effects on academic learning.

From the fifth grade research, we found most students value creativity and freedom in their assignments, as well as a sense of ownership and choice. However, we also found that students who felt they were struggling preferred the non-creative assignments. A solution to this disparity is limited or contained creative assignments. One way to achieve this is through multiple choice design options. In addition, we found creative treatments had no impact on efficacy and performance.

The second half of this thesis involved designing and launching a new creative class and evaluating its initial success. The course was taught to incarcerated youth at SLO County's Juvenile Hall. Given that the incarcerated youth population has very limited access to computer science education and that this education has benefits for this population, we believe it is important to introduce and study the effectiveness of this computing curriculum. In particular, we wanted to validate whether the creativity of game design in CS curriculum was engaging and empowering for the students.

In the end, we found our introductory course to be a positive experience for the students and staff. The students reported liking the class and an interest in continuing to learn computer programming. They also reported liking the game design aspect of the course and indicated they wished the course was longer.

Chapter 2

BACKGROUND

Two focuses of this thesis are evaluating the effects of creativity in a K-12 CS curriculum and expanding the reach of CS curriculum, specifically to San Luis Obispo Juvenile Hall students. For the former, it is important to understand some definitions and ideas used within research in this area. Some important background information to be aware of includes definitions of ‘self-efficacy’ and the overall research on creativity in education. For the latter, it is important to understand current research on computer science curriculum targeted at engaging high school students.

2.1 Self-Efficacy

Self-efficacy is a popular phrase and an important concept to evaluate whenever working within educational psychology [47]. When creating new curricula or testing ideas within education, it is valuable to see the effects of such on teacher or student self-efficacy, as seen in a plethora of papers in education research [40, 45, 46, 53, 54, 65, 75]. Since much of our work explores the effects of creativity and curriculum on students, it is nearly necessary to also evaluate student self-efficacy; thus, it is important to both define and understand this concept and to explain how to properly test for it.

2.1.1 Defining Self-Efficacy

Self-efficacy is a much debated term and defined in many different ways; furthermore, it is often confused with similar terms such as ‘self-confidence’ and ‘self-beliefs’ [47]. While there are debated views, the most cohesive definition, and most useful for the sake of our research, is as follows: “Perceived self-efficacy is defined as people’s beliefs about their capabilities to produce designated levels of performance that exercise influence over events that affect their lives” [17]. An important note is that it does not mean peoples’ actual capabilities, but merely their *beliefs* about their capabilities. As we will see, however, beliefs go a long way in determining results.

2.1.2 Importance of Self-Efficacy

According to Bandura, the primary researcher on self-efficacy, “self-efficacy beliefs determine how people feel, think, motivate themselves and behave” [17]. Research supports Bandura’s claim that it is not *just* a measure of how students feel, but that it also causes behavioral changes, impacts grades, and influences school retention rates [13] [18]. One study found it was a “uniformly accurate predictor of performance on tasks of varying difficulty with different threats regardless of whether the changes in self-efficacy were produced through enactive mastery or by vicarious experience alone” [15]. Therefore, it is a very important component to analyze with respect to making a successful curriculum; as such an important component, measuring efficacy should be done carefully.

2.1.3 Testing for Self-Efficacy

It is important to be sure of what you are actually testing - if surveys are not carefully crafted, findings may falsely attribute variables to affecting self-efficacy. There are many guides on how to properly test for self-efficacy. First of all, surveys must be testing on efficacy within narrow fields [14]. It is not practical or possible to test how one curriculum or experiment affects all realms of a student's efficacy - hopefully, our curriculum will positively impact student *computer programming* efficacy, but it probably should not affect athletic efficacy or public speaking efficacy. Therefore, it is important to write questions to test the specific domain the research focuses in, leading to the second key part of testing for efficacy.

Since it is impossible to have pre-made and pre-validated surveys for all possible fields of study, it is important to carefully craft and validate questions for each new domain researchers hope to test efficacy in. Bandura gives guides on how to create surveys for various domains [14].

Finally, it is also important to consider the audience - efficacy questions for adults are very different than those for children and adolescents. Bandura provides suggestions in this area of survey creation as well; in addition, there are also numerous sample surveys and tests created to test child self-efficacy in general and across many domains of research to consider and model and modify [26, 30, 33, 35, 45, 49, 60, 77]. There are also many tests of computer programming self-efficacy [21, 42, 73]. For our research, the challenge was modelling off of these two categories, and also narrowing the domain to an even more specific category based on the programming language and CS concept we specifically chose to teach.

2.2 Creativity

Another important topic of research for our study was that of creativity. Creativity can be defined in many ways. Some new research focuses on elements of defiance [66], while others define creativity colloquially for education, focusing on engagement, motivation, or flexibility in choosing learning tasks' [63]. Another fairly straightforward definition of creativity explains it as producing novel or original work that is appropriate and effective for the situation [39, 76]. While there are many differing explanations or definitions, the important converging point is that research has been narrowing in on the importance of creativity for society and thus its need within K-12 education [38, 39, 76].

2.2.1 Creativity in CS Education

Creativity in the context of CS education has some interesting nuances to explore.

First, researchers have begun exploring how *limited* creativity within computer science education can still pass as creative in the minds of the students [63]. Other writings explore how constraints are not only acceptable, but required for the creative process [67]. Our research explores this space, looking at the degrees of creativity that are most useful and engaging for students.

Another interesting area of research is how creativity can be used to break stereotypes within CS, making the field more inclusive. For example, work at the Georgia Institute of Technology looked at the gaps in perception between young adults and college students pursuing a career in CS. The young adults perceived computer science to be anti-social, boring, and not applicable to the real world (Figure 5), whereas the college students felt quite the opposite. The study suggested creativity as one of the

ways to engage more young adults in CS [74]. As discussed in the introduction, by reaching younger students with an engaging CS curriculum, in this case by using more creativity in computing curriculum, stereotypes about computing can be changed to create a more welcoming field of study for a wider audience.

Finally, creativity is an essential part of engineering in general, and thus should already be incorporated into curricula, despite the other aforementioned benefits. Teaching creatively makes better engineers, as innovation and novelty is an essential part of engineering by nature [27, 59]. Therefore, not only is it important for engagement, inclusivity, and breaking stereotypes - it is inherent to the field and thus should be inherent in its curriculum.

Reasons for Lack of Interest in CS (n=13)

Boring	9
Only for the Smart Students	7
Antisocial	5
Lacking Creativity	4
Tedious	3

Figure 5: Teenage Perceptions of CS [74]

2.3 Designing a Curriculum to Engage Middle and High School Students

Although, as mentioned above, creativity is one of the main focuses of our research, there are also many other techniques to keep in mind while designing an introductory curriculum. The second half of this thesis was designing and teaching an introductory CS course for Juvenile Hall students. To do this effectively, we explored ideas on teaching CS to high school students and on how to construct CS0 courses and courses for non-majors; research on engaging new students and students who did not choose a CS major could easily apply to our Juvenile Hall students as well, as there are many

overlapping needs. Overall, we explored and borrowed from research about teaching CS to young adults and the diversity of curricula needed to teach non-majors.

Research shows that young adults often believe the many CS stereotypes they have been presented with. One study found young adults believing CS was boring, only for the ‘smart’ students, antisocial, lacking creativity, and tedious [74]. Another study explored how adolescent girls often avoid CS due to stereotypes saying they do not belong, and found that one way to begin breaking stereotypes is through having non-stereotypical environments and classroom settings [22, 48]. A large part in creating equity is to break stereotypes before college; therefore, it is important to investigate what the preexisting stereotypes are, what perpetuates them, and how to create more engaging experiences for all students.

2.3.1 Curricular Approaches

There is currently a lot of research on how to make more engaging curricula for non-majors and young adults.

One technique for engaging non-major students is a project-centered curriculum. One study, which created a CS0.5 course for non-majors, discussed curricular approaches to meet the needs of the diverse group of students who were not familiar with computer science, nor necessarily going to pursue computer science after the first few courses. They found success in project-based courses with elements of student selection on their project topics [29]. Furthermore, other research indicates that project based learning increases self-efficacy [46]. One particular project that some students find highly motivating is game development. Research notes that games are fun projects which often allow students creative liberties [74].

Another idea to keep in mind while creating a curriculum is the type of goals for the

students. One study found that, for students learning math, proximal subgoals (close, attainable subgoals) over distal goals (larger, further away goals) caused intrinsic interest and personal efficacy [16]. This led us to strongly consider how to give students small subgoals along the way that would lead to the overall class goal.

Finally, we wanted to keep in mind how to best reconcile the desires for both breadth and depth in introductory courses. One study, which looked into the experiences of students and teachers participating in K-12 curricula, found “a preference for a hybrid approach over pure skill-deepening or contextual scaffolding” [64]. Another study creating a CS0 course found teaching with JavaScript as an effective way to balance depth and breadth [58]. They report that its “simplicity, natural interfaces, and seamless integration into Web pages make it possible for novices to develop interesting and engaging programs quickly” [58]. Other studies have found similar benefits through using Processing, a Java-based language for 2D graphics [71, 72].

Chapter 3

RELATED WORK

Numerous groups have begun striving for equity in CS education, and many have also focused on the K-12 space. Various techniques have been attempted, some more successful than others. Some notable and inspirational efforts to create equity in CS are outlined below.

3.1 Code.org

Code.org is a nonprofit group that offers free online CS courses to K-12 students. They utilize block-based coding and online modules to try to expand CS to students who previously had little or no access to it. They have done notable work on not only identifying and reporting on gender, racial, and socioeconomic gaps in CS, but also on closing this gap [6]. They describe themselves as “dedicated to expanding access to computer science in schools and increasing participation by women and underrepresented minorities” [5]. This is seen in the make-up of their classes - Code.org has incredible reach within their classes as opposed to, for example, the AP CS spread. They report 46 percent female students, 48 percent underrepresented minorities, and 47 percent students in high needs schools [5]. They recognize that it does not only matter who the material is made available to, but how it is represented: it must be done in a way to foster interest for the target audiences. This organization also has done work in spreading CS globally, offering courses in over 63 languages and in over 180 countries, and even in doing work to change policy to increase CS participation [5]. Overall this group has made a large impact in the K-12 space.

Code.org offers an amazing model of spreading CS education to young students. That said, there are distinct ways our work differs from Code.org curriculum. Code.org uses online modules, which are not accessible to all students - for example, online modules are sometimes useless in settings such as Juvenile Hall, where internet use is strictly monitored. Our work contributes to this area by creating a curriculum for these settings. In addition, Code.org (and many similar programs) use block-based programming for younger students, where code chunks are dragged and dropped. (Though Code.org *does* offer options to use JavaScript in addition to block-based coding for some assignments for older students, most of the modules, especially for younger students, are exclusively block-based.) While this does provide simplification to the complex confusion of learning to code, our curriculum purposely uses text-based languages so that students will have a smoother transition to future CS courses [61]. Another way our work is differs is that it is not just teaching and creating a curriculum, but is also experimental; it focuses on discovering how creative choice within the curriculum affects learning. A lot of online modules miss aspects of creativity in the generality needed to reach all sorts of students and in the goal of simplicity (though there *are* some creative assignments offered as well in addition to the more restricted modules). We explore student reactions to these varying degrees of creativity.

3.2 Bootstrap

Bootstrap is a group that also offers free, online introductory computer science courses to middle and high school students. They work with school districts across America, offering workshops for teachers and courses for students. Bootstrap also does amazing work in creating equity in CS education. Their curriculum is research focused, based out of Brown University [3]. Their goals are equity, scale, and rigor and they are proponents of CS for all. They strive to reach women and underserved minorities,

serving nearly 50 percent African American or Latinx students and nearly 45 percent women. Their curriculum is for 6-12th grade students, and one of its key features is that it teaches coding concepts through algebra (and also physics and data science). This way, the curriculum can be taught as a standalone course or can be integrated into AP CS, math, or science courses. Not all high schools currently teach CS, but all must teach algebra - this integration of material allows for further accessibility. Bootstrap also addresses some of the concerns we had with Code.org. It is text-based, using the functional language Racket. They also provide online and offline options for using their materials.

Schanzer et al. has worked to validate Bootstrap’s algebra course and creativity within its curriculum [62, 63]. Their research shows that student math skills do in fact improve through their curriculum, indicating the success of this integrated approach. Their research also explores creativity withing the curriculum. Their findings are supported by our experimental findings, as questions such as “what kind of personalizations do students make?” and “How many degrees of freedom do students need to feel a sense of ownership?” were mirrored in our own study. In addition, our research furthers this work by asking more detailed questions - due to the constraints of using only one school, we were able to have ‘before’ and ‘after’ surveys as opposed to just one survey asking students to recall how they felt at the beginning of the course.

Another difference in the creativity research is that Schanzer et al. tries to identify if having only four customizable pieces is enough to create a sense of ownership. (Students making a game were allowed to design characters, the static background, items collected, and items to be avoided). Our study differs because in our fifth grade class, the students were constrained in an opposite manner. Instead of being given full flexibility on only four aspects of a game, we gave them only four options on many

aspects of a drawing. Therefore, our work explores a similar question with a different twist. In addition, the game our Juvenile Hall students made during our course is similar to their model, but, once again, the options vary. For the Bootstrap game, students can select characters from online images, which is seen as creative and free. Our students could *make* any character from shapes and lines, which is possibly more creative since students can make anything, but also more limiting since it is slow and difficult to make complex characters from scratch. It would be interesting to compare student engagement with those options in the future. In addition, our study differed in that, due to the very small class size, the students were allowed to make nearly any edits they could think of to their games, adding immensely to the creative options. Schanzer’s paper posed the question: would students feel less creative ownership when comparing this structured game creation to more free-form creation? Our curriculum was much more free-form. In the future, it would be interesting to take their question and compare Bootstrap to our curriculum across various settings.

Another difference between the work of Schanzer et al. and our research is on breadth of research. Schanzer et al. had a broader sampling of students (225 students across multiple schools versus around 100 students at one school) and also had a simpler intervention. Our work focused less on making scalable material (which is Bootstrap’s goal) and more on making better material for our specific students. With a smaller sample size, we were able to ask more questions, distribute more surveys, and have more interventions.

3.3 MyCS

MyCS is an attempt by Harvey Mudd research to combine all of the many CS education tools/curriculums targeted for 4-10th graders into a cohesive curriculum [64].

Their work focuses on teachers and partnering with them to meet specific needs. They put an emphasis on fostering a “computational identity” and breaking stereotypes by teaching that “CS is something that people like me do.” Similar to Code.org, their curriculum is based on the block-based language Scratch. Their work with teachers validated the need for breadth and depth when teaching CS and also reported the need for slowing down the curriculum. They also reported on increased self-efficacy from their curriculum. Our work differs from this work by doing less of a breadth analysis on teachers and focusing more individually on the two teachers we were working with. In addition, the self-efficacy portion of their research was possibly mislabeled - no validation was given on how they showed they were actually testing self-efficacy. We attempted to provide some validation in our research on if we were actually testing for efficacy or not. In addition, while we tried to pull from the teaching suggestions this research shared, we still were using our own curriculum with offline options.

3.4 Peabody Charter Outreach

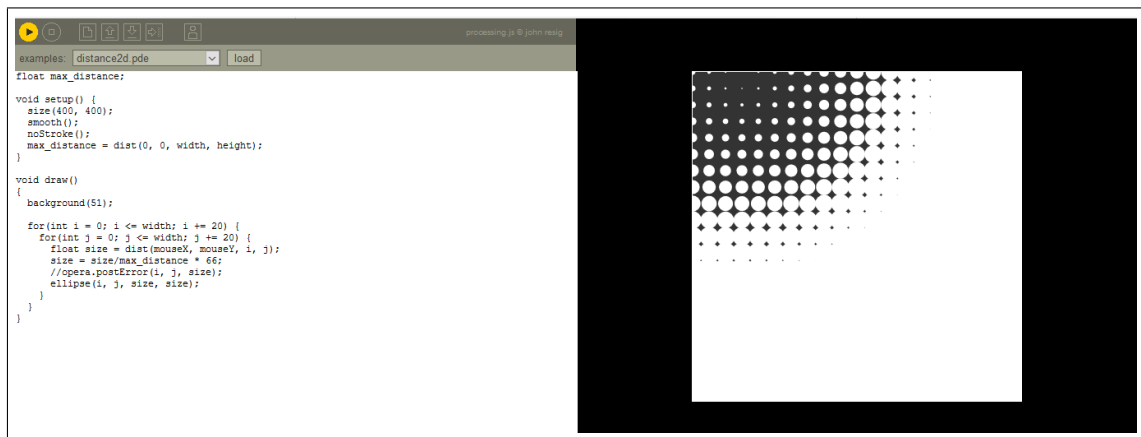


Figure 6: Online Processing Editor [9]

Since 2014, Peabody Charter Elementary School has been holding computer science labs for fifth grade students there. The curriculum is designed to teach basic com-

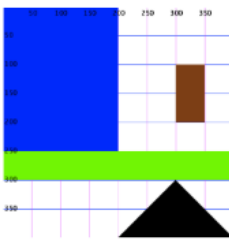
puter science through 2D computational art. The students code in Processing, an open source, text based programming language based on Java, using an online editor (shown in Figures 6) [1, 71]. The curriculum was designed by Cal Poly Professor Zoë Wood, fifth grade teacher John Wilcox, and a Cal Poly master’s student Katie Davis, and it teaches concepts through 2D puzzles and creative activities [28]. The students are taught concepts as a group and then are guided through exercises on their own laptops with paper worksheets for guidance (Figure 7 shows a sample of one of these worksheets). This lab is held for an hour once a week for the duration of the school year.

Name: _____

Processing: More puzzle games!

Z. Wood & J. Wilcox & K. Davis

Today we are expanding on last week's puzzles programming! Our goal is to practice moving and creating ellipses, rectangles, and triangles to create pictures. We will start with the same 400 by 400 grid as last week with some extra shapes already drawn for you.



Starting Code

```

size(400, 400);
background(255);
//the horizontal lines
stroke(0, 0, 255);
line(0, 50, width, 50);
line(0, 100, width, 100);
line(0, 150, width, 150);
line(0, 200, width, 200);
line(0, 250, width, 250);
line(0, 300, width, 300);
line(0, 350, width, 350);
//the vertical lines
stroke(255, 0, 255);
line(50, 0, 50, height);
line(100, 0, 100, height);
line(150, 0, 150, height);
line(200, 0, 200, height);
line(250, 0, 250, height);

```

```

line(300, 0, 300, height);
line(350, 0, 350, height);
stroke(255, 255, 255);
//house
fill(0, 100, 200);
rect(0, 0, 200, 250);
//roof
fill(0, 0, 0);
triangle(200,400,300,300, 400,400);
//grass
fill(0, 255, 0);
rect(0, 250, 400, 300);
//door
fill(150, 75, 0);
rect(200, 100, 50, 100);

```

TODO: Change the numbers in the starting code to move the shapes and recreate the puzzle 1 below. Add new shapes to create puzzles 2 and 3. For each of these puzzles you need to think about how to make the shape draw where you want it to go (use the grid to help you!) Recall that the shape commands are:

- `ellipse(x, y, width, height);`
- `rect(x, y, width, height);`
- `triangle(x1,y1, x2,y2, x3,y3);`

Puzzle 1 (intro level):
This puzzle requires you to take the 4 shapes drawn from the starting code and change the numbers so that the picture matches the one below.
Suggestion - Starting with the grass and building up, move one shape at a time until it fits into the picture.

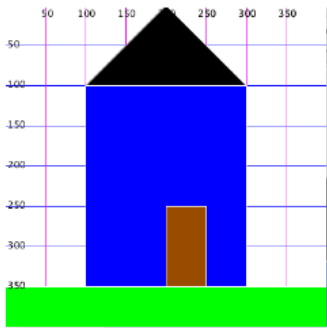


Figure 7: Sample Curriculum [71]

This thesis directly builds off of the previous Peabody Charter labs. Four sets of worksheets, modelled strongly off of the preexisting curriculum, were designed to specifically test certain aspects of how creative choice affects student preference, self-efficacy, and learning. These interventions were taught during six of the usually scheduled lab weeks, and were planned with the original professor and fifth grade

teacher who first designed the curriculum.

3.5 Processing Classes

The work of Dianna Xu et al. reports on turning a CS1 computational art course into a high school course launched at two very different schools [72]. The curriculum was designed using Processing, which is what our curriculum is built on as well. Their work shows an increase in student engagement, enrollment, and female participation as well [72].

Zoë Wood had already created and adapted a computational art curriculum using Processing for fifth grade, high school, and a college level CS1 course [71]. This is what our research is pulled from and modelled on, specifically the fifth grade portion. However, Dianna Xu et al. provides further validation for using Processing with a diverse range of high school students, which is what we aimed to further do by also expanding this style of course to Juvenile Hall high school students. Our work also differed from *both* models of computational art by creating a Processing-based game design course for the Juvenile Hall students instead of an art course.

3.6 Introductory Computer Science Courses with Restorative Partners

Restorative Partners is an organization in San Luis Obispo that educates the community on and practices restorative justice. They look to minister to those at San Luis Obispo County’s Juvenile Hall, San Luis Obispo County’s Jail, and the County Jail’s Honor Farm, providing a “continuum of care from incarceration through reincorporation” to the community [2]. One of the ways Restorative Partners provides restorative justice is through offering a wide variety of classes. These classes engage

the community by bringing in individuals with a broad range of skills to teach classes at SLO County’s Juvenile Hall and Jail, fostering learning and community.

Cal Poly professor Theresa Migler, teamed up with multiple students and Restorative Partners, created and launched an introductory computer programming course in Python. She teaches at Women’s County Jail and Men’s Honor Farm [50]. In order to teach in these environments, she specifically designed the curriculum within certain guidelines. For example, the student computers could not have access to WiFi during the class and computers were only available to the students *during* the class. This led to a careful curriculum, where all practice and homework assignments were done by hand - this way learning could continue outside the classroom, but still meet the specific needs of the situation.

This program aligns with the goals of this thesis - extending computer science courses to individuals who previously did not have access and specifically designing the curriculum for individual needs of the group. How this thesis differs is in extending classes to a *younger* population. As mentioned previously, reaching K-12 age is of utmost importance in creating equity. By partnering with the previous work done at the San Luis Obispo County Jail [2, 50], and creating a curriculum specifically for this younger audience, we can spread computer science and empower this demographic.

Chapter 4

FIFTH GRADE CURRICULUM

This portion of the the thesis evaluates the effects of creative choice in a fifth grade computer science curriculum.

Demographics

We conducted our research at Peabody Charter Elementary School in Santa Barbara (see Figures 8 and 9 for this school’s demographics). As mentioned in Chapter 3, four fifth grade classrooms (consisting of 107 students) attended a computer science lab for an hour once a week. One of the classes had 26 students, and the rest each had 27 students.

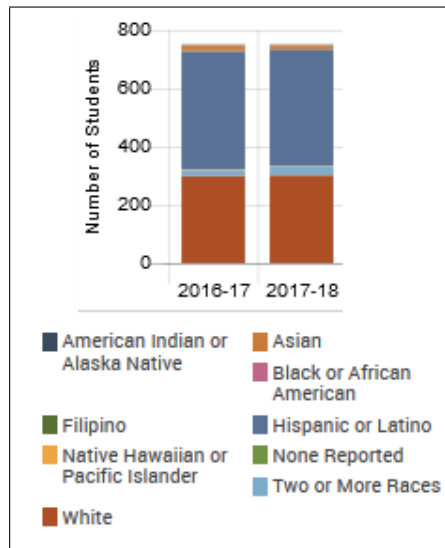


Figure 8: Student Enrollment at Peabody Charter by Ethnicity.

Lab

During lab, students worked on 2D computational art, coding in the language Processing. They used a curriculum designed by Zoë Wood and a fifth grade teacher at

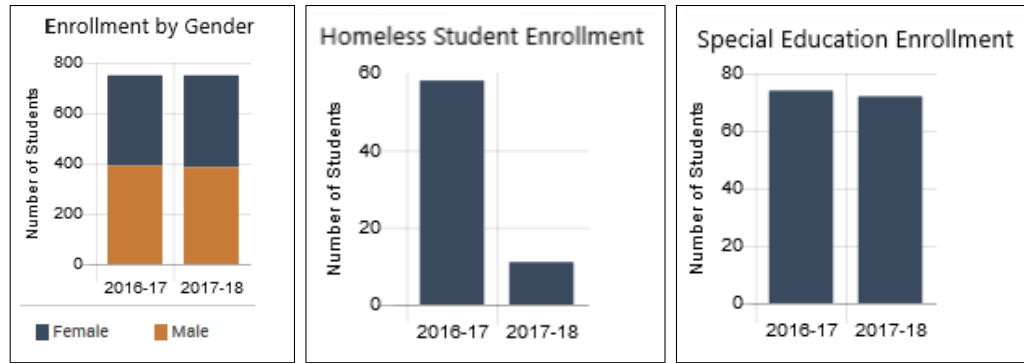


Figure 9: Student Enrollment at Peabody Charter by Other Demographics.

the school, John Wilcox, who also led the computer lab each week. The students received a 10-20 minute lecture each session consisting of demos and filling out questions on a worksheet together. Afterwards, they worked on finishing the worksheets on their own.

Our contribution consisted of two interventions in this lab, testing different effects of creativity on the students' views, learning, and efficacy. We designed our own worksheets, modelled off of the worksheets the students were used to, in order to test varying degrees of creative choice. We also distributed google form surveys at different points to gather data.

Research Approval

In order to conduct research with children, we applied for and received research approval from Cal Poly's Institutional Review Board (IRB), along with the school's permission and parent/guardian permission for each student. The IRB proposal, the permission forms, and scripts are included in Appendix A.

Motivation

The motivation of our research was to discover if creativity and giving students choice can increase engagement with CS curriculum in a wider range of fifth grade students.

As mentioned in the Chapter 1, when the fifth grade students were given a survey at the beginning of the school year, only 40.9 percent of students reported enjoying math and 63.6 percent of students reported interest in computer programming. However, 83.7 percent of students reported enjoying art. Therefore, one hypothesized way to increase student interest in programming is to have art applications.

The fifth grade curriculum already had art incorporated, as it teaches coding through 2D computational art. However, one of the key aspects of 2D art is not just the visual display of images, but the process of creating, designing, and choosing what images to produce. Computer programming, however, is a complex subject for fifth grade students to grasp. Although the visual aspect of art was always incorporated in the fifth grade lessons, it is often difficult to give students full creative choice over the art produced. Giving students choice can often increase complexity when they are already confused - often, restrictive instructions seem necessary.

Therefore, due to student preferences for art, but also the complexities of learning a new subject, we desired to explore the nuances of student opinions in this area. Did students reporting an enjoyment of art also mean they would enjoy creative choice in coding assignments? If so, how much choice is best? While learning difficult material, would students actually choose more complex and creative options or simpler paths? These are some of the questions we explore in the two fifth grade interventions.

4.1 Intervention One

The first intervention focuses on analyzing student actions and reported preferences concerning creative choice while reviewing/working with material they already knew. The focus was to evaluate student opinions on creativity with as little external influences as possible.

4.1.1 Design

Worksheet Design

Overview

The intervention was a combination of two worksheets (for two lab periods) and a survey. Both worksheets had students write the code to draw a simple picture of a character using Processing. One of the two worksheets contained varying possibilities for creative choice. While constructing a picture, students were given multiple choice options on how to make the different features (shapes and colors) of the character they were drawing. For each feature, they could pick from three provided options or pick to create their own feature. Option ‘D’ was always an option for the students to pick “other” and design their own feature if they did not like our suggestions, as seen in Figure 10. The second worksheet involved the students typing specific instructions step-by-step (no creative choice allowed), as seen in Figure 11. The former worksheet is referred to as the creative worksheet, CC worksheet, or students having “options”, whereas the latter is the non-creative, no-CC worksheet, or students following “steps.

►Next, let's pick a background color to place our snowman in. Pick from the following or make up your own. Find the line "background(255);" in the code you pasted and replace it with your choice.




A. Light Blue  background(0, 200, 250);	B. Dark Blue  background(17, 30, 108);
C. Light Purple  background(150, 111, 214);	D. Other background(____,____,____);

Figure 10: The First Instruction for Selecting the Background Color for the Creative Choice Worksheet.

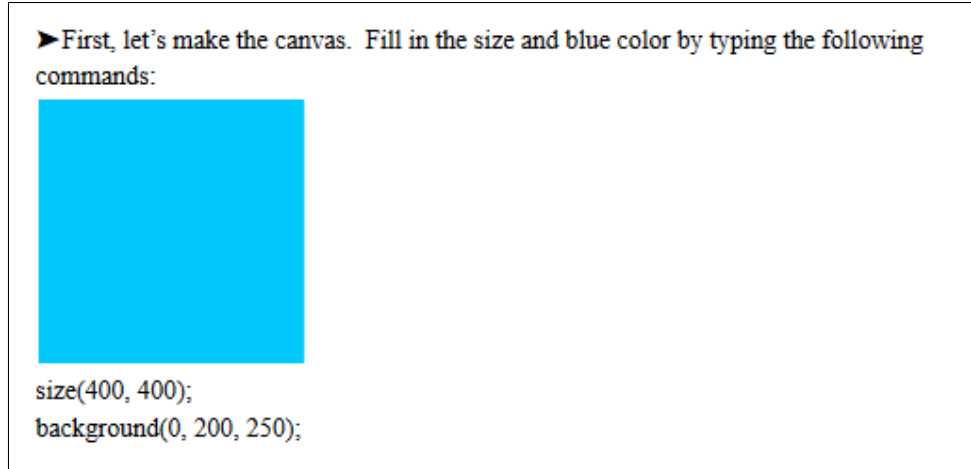


Figure 11: The First Instruction for Making the Background Color for the Non-Creative Worksheet.

Constraints: Reducing Complexity

In order to reduce the number of variables, the worksheets for our first intervention were not designed to be very challenging - they were merely guided steps showing students what they could do with the commands they had learned thus far. We reserved more complex assignments for *Intervention Two*, so this one could focus on isolating creativity as much as possible. One of the fears of making more complex assignments was that if students were stuck on a problem on one of the two worksheets for whatever reason, they might automatically like the other more. Therefore, we chose to give them base code for the no-CC and the CC assignment (aside from the “other” option students could choose) to try and isolate creative choice as the only difference instead of difficulty.

Constraints: Giving Each Treatment to Every Student

For our first intervention, we wanted to gather student opinions after giving each student both assignments to choose between, as opposed to comparing class averages after assigning one treatment to each class. While there is value in looking at trends in classes as a whole (and we did do this in the second intervention) there are also issues

in this approach, as each class is different in student composition and abilities; we wanted to see what a student would prefer when given two assignments (one creative and one not). Due to this desire, we chose to have the worksheets based on reviewing material they had previously learned instead of teaching new material (saving this also for the second intervention). If we taught new material to the students we could not apply both treatments since students would already know the material by the time they were given treatment number two. For this reason, and other simplifying effects, we chose to give two worksheets which were as similar as possible, working with identical concepts, aside from the creative choice.

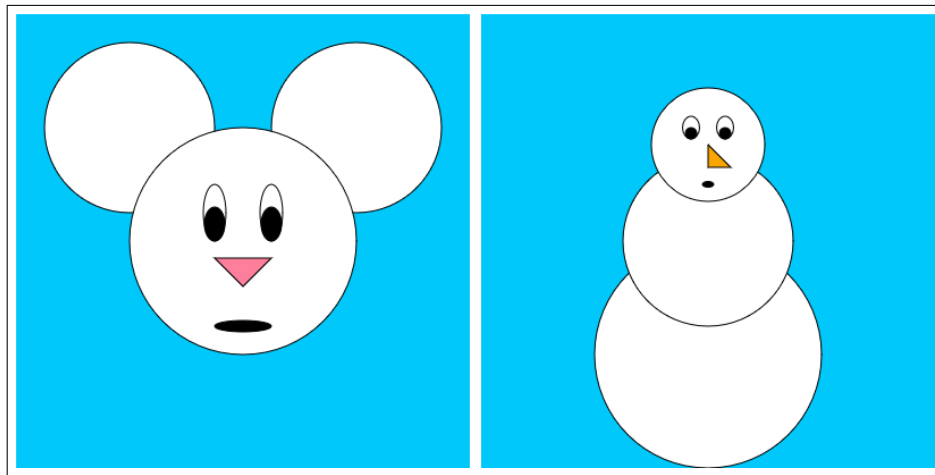


Figure 12: The Two Images Made by Each Student During *Intervention One* (Though One of the Images Varied Depending on Which One Was on the Creative Choice Worksheet for a Given Student).

Since each student was to get both treatments, the worksheets *did* need to have more variables than merely creativity. We had to have students making different pictures each week to prevent boredom and dislike of the second worksheet. We chose to have students make a mouse one week and a snowman the other (Figure 12). To help mitigate the necessary variables, we made sure each picture had the same number of shapes and color selections. We also made sure, out of the four classes, that two were given a snowman as the CC assignment and mouse as the no-CC, and two classes were

given the mouse as the CC and a snowman as the no-CC assignment. We also made sure that two of the classes were given the CC assignment first and the other two classes were given the no-CC assignment first in case students were biased to prefer the most recent assignment they completed. Each worksheet took one lab period, so they completed one worksheet the first week and the second worksheet, followed by the survey, the second week. By making sure the classes were flipped in which type of worksheet they were given first, we could minimize skews due to students preferring/only remember the most recent assignments. We also were sure to isolate *why* students chose worksheet preferences to help mitigate the variables introduced by provided different images (a mouse and a snowman) each week. We were sure to evaluate if students made preference choices based on ‘unrelated aesthetics’ or due to creativity levels, which is discussed further in survey design.

References

The worksheets for the creative choice mouse and snowman and the non-creative choice mouse and snowman can be found in Appendices C, D, E, and F respectively.

Survey Design

The survey was broken into two portions. The first half of the survey asked students which worksheet they preferred in general, not specifically asking about the creative choice aspect of the worksheet. After making their selection, they were asked to check reasons for which they preferred that worksheet from a predefined list of options. This was so we could analyze how important creativity versus design or superfluous reasons were for liking an assignment. We opted to give checkboxes, despite the risk of implanting ideas or skewing student thinking, because sometimes the fifth graders needed prompting to answer questions. We also had them say which was the most important reason they checked, and also gave them an “other” option where they could write whatever they wanted in order to try to offset the rigidity of checkboxes.

The second half of the survey asked students specifically if they preferred having options more or following steps more, regardless of which worksheet they preferred as a whole. They were given space for a free-response explanation as to why they had this preference. The free-response question was to see the self-reported descriptions of their preferences without skewing or biasing them with predefined options.

The survey can be found in Appendix G.

4.1.2 Research Questions - Design, Results, and Threats to Validity

Research Question Concerning Overall Worksheet Preferences

These questions examine the first half of the survey, where students were asked their overall worksheet preferences and why. This was to see if students picked preferences based on the creativity differences or for other reasons to determine how important creativity and design choices truly are to our students.

Research Question 1: Which worksheet did the students prefer out of the two they were given?

Research Question 2: What are student reasons for their worksheet preferences? For those that prefer creative assignments, do they actually say it is because of creativity? Why do students prefer less creative assignments?

Design

Students were asked which worksheet they preferred. While one worksheet had choice and one did not, they were not asked to choose their preference based on this. As mentioned above, they were given a list of reasons to check and an opportunity to write their own reason.

This section is particularly interesting because it helps reveal what students find to be more important. It narrows down if the students identify creative choice as an important reason for liking an assignment, or if, when not specifically asked about it, they find other unrelated aesthetics more important. It helps focus how important creative choice is to them and if it is strong enough to sway preference.

Results

There were eight options the students could select for why they picked the worksheet they picked. Those answers could be clustered into the categories shown in Figure 13, which include: creative/freedom, easier/completed, challenge/comprehension, and unrelated.

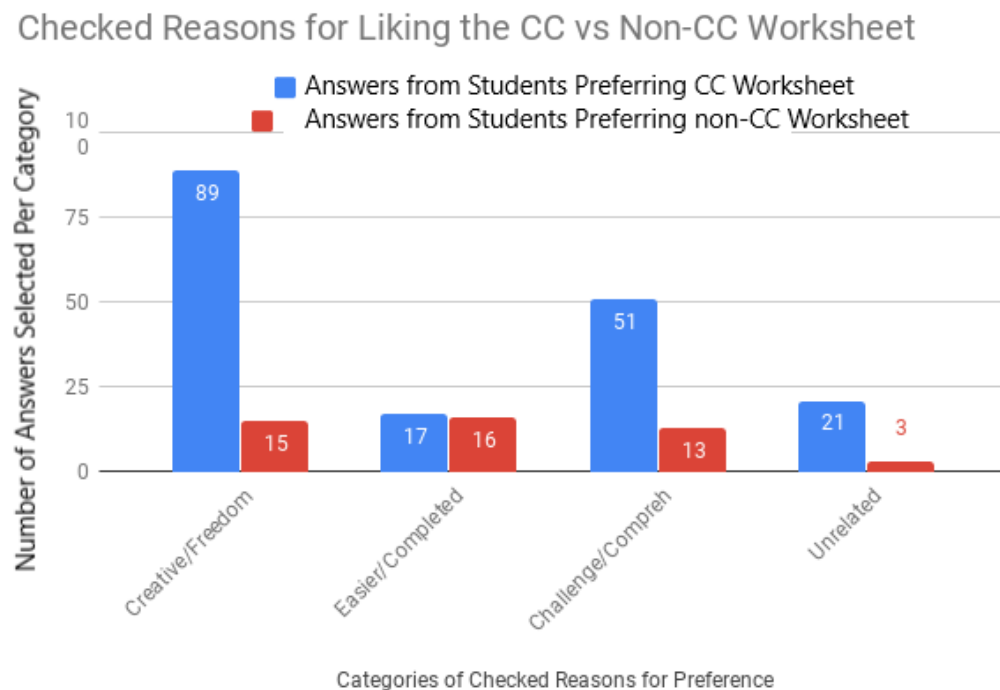


Figure 13: Checked Reasons for Worksheet Preference

Our hypothesis was that most students would prefer the creative assignment. We also predicted that most preferences would be due to the creativity or lack thereof

over unrelated aesthetics, with students who liked the creative assignment reporting it was because of the creativity while the minority of students who did not would report they were struggling with the material.

The initial results support our hypothesis, as 78.75 percent of students *did* prefer the CC worksheet. Furthermore, the most common reason for preferring the CC worksheet was creativity or freedom (see Figure 13). The most common reason for preferring the non-CC worksheet was it was easier or they completed it (see Figure 13). However, a close second was tied to the creative/freedom category. This was not as expected, and does complicate the results. However, it could be a result of students misinterpreting freedom to mean they finished faster and got more free-time (which some of the free response answers lead us to believe).

Only 41.25 percent of students chose to fill out the “other” option for why they picked the worksheet they preferred. For those who did, their free response answers are grouped into categories. The count of students whose responses fell into the categories (unrelated aesthetics or benefits, easier/less confusing, learned better, challenge, fun, freedom, choice/picked/ownership, creative/designed/made options, and unique appearance) are shown in Figure 14.

The most common “other” reason students preferred the CC worksheet had to do with choice, picking their options or appearance, and ownership of their work. This helps support our hypothesis of students preferring creative choice in CS curriculum. The results are strengthened by the fact that students could have picked preference for any reason (as seen by the second most popular choice being Unrelated Aesthetics or Benefits - e.g. snowmen are winter themed), yet still most commonly picked CC and picked it for the aforementioned reason.

Also of note, when counting all categories that had to do with not only the words

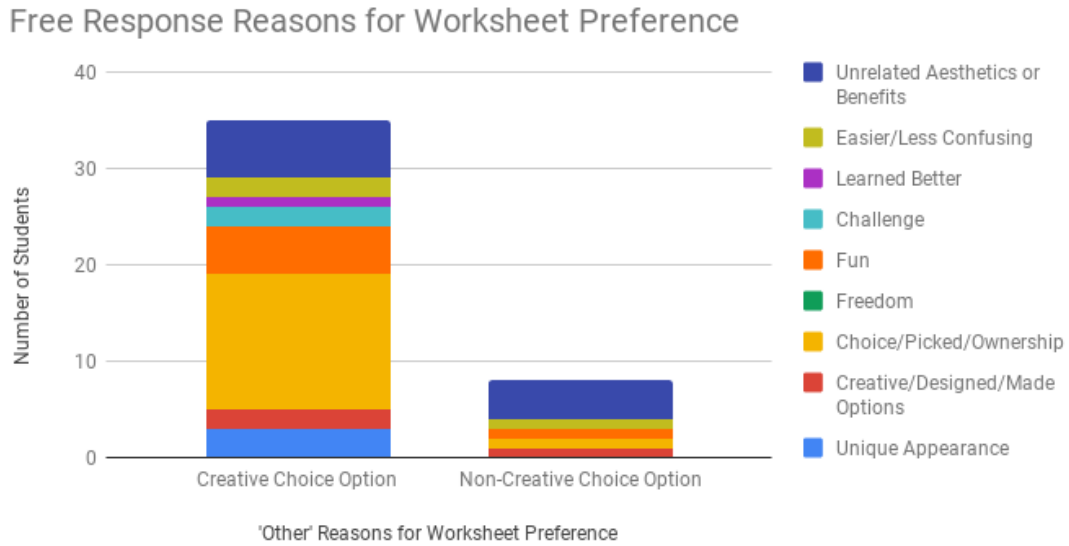


Figure 14: Free Response for Worksheet Preference

‘choice’, but also ‘creative’, ‘design’, ‘freedom’, or ‘unique appearance’, more than half of the free-responses support the hypothesis.

Threats to Validity

As mentioned above, the results were probably skewed at least partially by the language used throughout the experiment and also by the provided checkbox options. Some students might have interpreted ‘freedom’ as free time if they finished an assignment early (although we did instruct them to only answer about the worksheets and not about anything else like what they did once they finished). We also took this into account during the second intervention and were sure to apply the survey *as soon* as they finished the assignments to be sure they did not associate free time with the worksheets. We attempted to mitigate the checkbox skews by providing an “other” option and by also providing a free-response question.

Research Question Concerning Creative Preferences

These questions look into the second half of the survey, where students were asked directly if they preferred the creative choice or following steps better. This was to see, in case students picked their overall favorite worksheet based on unrelated aesthetics, what they actually thought about what we were trying to test.

Research Question 3: For students that claim to prefer creativity, regardless of overall worksheet preference, how do they define it? What aspects of “creativity” do they like or think of when hearing this word?

Research Question 4: What are student reasons for claiming to like non-creative assignments?

Design

The second half of the survey asked the students directly if they preferred being given options or being told to follow steps, regardless of which worksheet they preferred overall. Although this skews student thinking by specifically telling them what the worksheets were looking at, it allowed students to decide which style they preferred regardless of whether it was the style of their preferred worksheet or not. For example, a student might have preferred the snowman worksheet because it was winter-themed and the experiment was conducted in the winter. However, even though that reasoning led them to prefer the snowman worksheet, they may have still preferred the creative choice style of the mouse worksheet, and would have liked a creative choice snowman best. Therefore, asking specifically about being given options or not also seemed important information to have.

Results

After reading all of the responses, the student answers were counted and grouped into

the following categories, as seen in Figure 15.

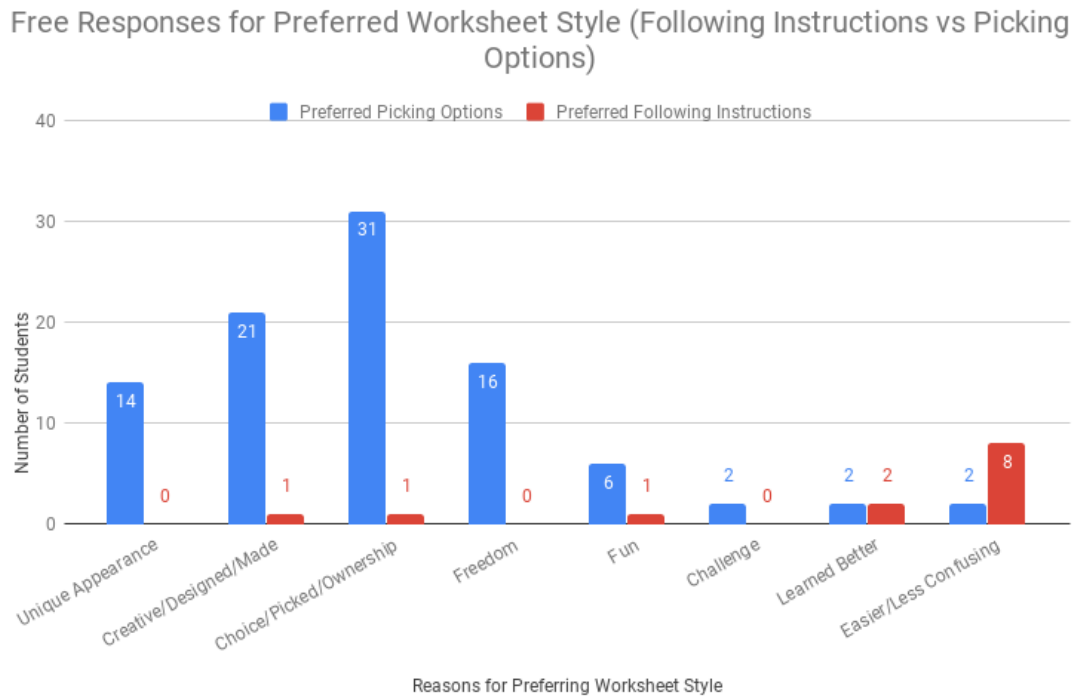


Figure 15: Free Response Worksheet Style Preferences (Following Instructions or Picking Options)

The most common reasons students gave for preferring ‘options’ were the ideas of ‘ownership’, ‘choices’, or ‘picking’. The second most common reasons were ‘design’ or ‘creativity’.

Examples of student responses binned in the top four categories are as follows:

- Choice/Picking/Ownership: *“I liked it better because I like making my own disition.”*
- Creative/Designed/Made: *“because I got to design and it felt less like a work-sheet.”*
- Freedom: *“I liked the amount of freedom I had”*

- Unique Appearance: *“beacause everybodys looked differnt”, and “because if we followed the instructions all of are makings look the same”*

The few students who preferred following steps liked it because it was easier or less confusing. As expected, some of the students that preferred following steps reported struggling with Processing thus far and appreciated the simplicity of not having choice, supporting our hypothesis. Some student responses exemplifying this were as follows (copied verbatim from student surveys):

- *“because then we get to know it better”*
- *“Because it helps me learn coding better so I can take notes and use them for other coding projects.”*
- *“i like better because i know what to do”*
- *“it felt easier to me”*
- *“beacuse then i can understand the meaning.”*
- *“because you will know what to do and no to get confuzed”*
- *“I liked this option better because I am not very good at processing yet.”*

One student reported the following, though most preferred following steps because of the above reasons or because it was easier:

- *“because it didn’t take long”*

Threats to Validity

As mentioned above, this part of the experiment risked skewing student thinking by

explaining the styles of worksheets. For example, students might say they preferred creativity because they think they are supposed to or it sounds better. However, research questions 1 and 2 addressed this possible issue by asking their opinions before explaining the experiment, mitigating this threat.

Research Question Concerning Worksheet Preference vs Creativity Preference

After analyzing both worksheet preferences and creativity preferences, we wanted to see how many students did not have matching responses. Did many students, despite liking the creativity on one worksheet still pick the other as their favorite? This section aims to further narrow in on how strong of an importance creativity holds in the process of student engagement with computational art.

Research Question 5: How many students still picked a worksheet as their favorite despite preferring the other worksheet's presence of or lack of creativity?

Design

As mentioned above, we asked students to report which was their favorite worksheet and also if they preferred to follow instructions or to be allowed to choose from options.

Results

Overall, most students (81.25 percent) matched their overall worksheet preference match with their preferred style of worksheet (creative choice or following instructions). This is a positive indication that creativity plays a large role in student engagement. Despite extraneous artistic preferences (preferring snowmen over mice or vice versa), *most* students seemed to pick preferences based on being given creative choice or not.

Only 18.75 percent of students reported their favorite worksheet as having their less favorite style of creativity. Of these students, 12.5 percent had chosen the less creative worksheet as their favorite despite preferring creative choice over following directions. Overall, this is a very small portion of students who preferred creative choice but picked their overall worksheet preference based on unrelated aesthetics. This indicates the importance of the creative choice aspect of art over merely 2D visuals in engaging students.

Threats to Validity

As with all the survey questions, there is a chance students misunderstood our questions or were skewed by the options presented in the checkboxes. However, they were asked to report their worksheet preference before they saw the checkbox options, so the threat is very small for this research question.

Research Question Concerning Degrees of Creativity

This portion of our research looks at how the students actually utilized the levels of freedom allowed. Would they choose scaffolding or full free-form design when given both options? We felt this was an important area to research because more freedom often creates curricular difficulties. Without some form of bounds, it is difficult to move a large group of students towards an end goal. If students felt they had choice without needing or wanting to go to the extra effort of designing elements from scratch, it could be helpful information in creating future curricula.

Research Question 6: Are students that report to prefer picking options over following instructions more likely to exercise their creativity?

Research Question 7: For students that claim to like creative choice, what percentage of them actually chose to create their own pictures when it involved more

work?

Design

The creative choice worksheet, as mentioned in the worksheet design, offered multiple levels of creative freedom. Students had three options (options A-C) of pre-designed features (colors and geometry), or could select to create their own feature from scratch (option D). For each feature, be it nose, head shape, or eyes, students could pick from the three options or pick to create their own. The only exception was, if for the overall shape the student picked to create their own shape, all subsequent design options were automatically counted as selecting option D (e.g. if the student designed their own snowman body type from scratch, then the eyes, nose, and mouth of the snowman were also automatically option D because there were no predesigned options for their made-up body type).

Results

With the rules defined in the previous section taken into account, we wanted to see how many times students actually chose to create their own option versus select pre-made ones. Our hypothesis was that students that reported to prefer getting choice would also have picked to design their own options with a higher frequency. Figure 16 shows that the average number of created options per student *was* higher for students that said they preferred picking over following instructions. However, the difference is very small. In addition, 51.51 percent of students that reported to like picking over following instructions actually chose to pick option ‘D’. This is higher than the 21.43 percent of students who liked following instructions that picked at least one option ‘D’.

It seems, from an initial review of results, that while most students reported to prefer having options, ownership, and different results from their peers, they rarely chose to

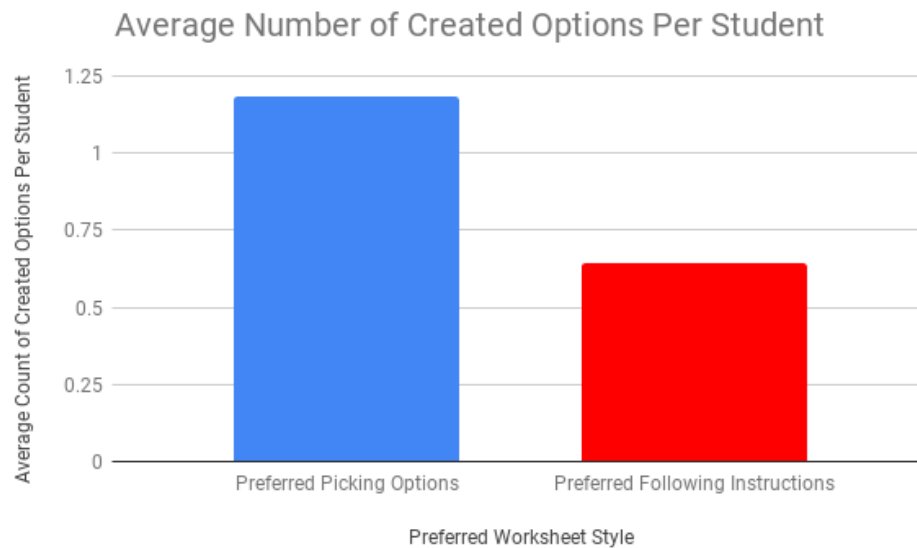


Figure 16: Average Count of Created Options by Worksheet Style Preference (Picking Options or Following Instructions).

exercise complete creativity. 53.75 percent of students never picked option ‘D’. Some new hypotheses from these initial results are that students want to feel they have choice, but still like some bounds and guidance on that choice. Or perhaps, students want to be given freedom and choice, but not at the cost of needing to do more work. Perhaps a limited freedom is the best approach to pursue in further research.

Threats to Validity

We found that many students reported to like creative choice, yet felt no need to exercise their full freedom by choosing option ‘D’. However, this does not necessarily mean students would be just as happy with the creative assignment if the option ‘D’ were removed. Similar to our study, Bootstrap research has found that students still feel the creative freedom with a limited degree of it; however, neither our research nor theirs experimentally compared student results when given differing degrees of creativity [63]. Both only found that students reported they enjoyed the amount of creativity even when they did not choose to use all of it, or reported they enjoyed the

amount of creativity when it was limited in a particular way, but those two versions of creativity have not been directly compared.

4.1.3 Summary

Overall, the first few research questions found that most students (78.75 percent) preferred the creative assignment over the non-creative assignment. They reported liking the assignment because of the creativity and freedom. The students that gave free response descriptions of why they liked the creative assignment reported liking having choice, getting to pick from options, and having ownership over their work. However, we also found that some students (21.25 percent) preferred the non-creative worksheet because they were able to complete it or because it was easier. However, the students who gave free response reasons as to why they preferred the non-creative worksheet were mostly because of unrelated aesthetics or benefits.

From the second set of research questions, we found the most common reason students reported to prefer picking from multiple choice options was a sense of ownership, having choice, and getting to pick. The students who preferred following instructions reported they liked that it was easier and less confusing.

We also observed how creative choice played a larger role in student engagement than other unrelated reasons.

Finally, from the last set of research questions, we analyzed if students would use the full amount of creative freedom offered to them. Only 46.25 percent of students chose to create an aspect of their image on their own. The rest of the students exclusively used the suggested options we provided.

Considering the results, we recommend assignments with limited amounts of creativ-

ity. We attempted to make our creative worksheet straightforward. However, as much as we tried to isolate creativity from worksheet complexity, there is inherently some complexity associated with creative freedom. Hence, a limited style of creative worksheets seems preferable. Furthermore, students reported a sense of ownership from our multiple choice assignment, but very few students actually chose to pick option ‘D’, the create-your-own option. We hypothesize that we could simplify the assignment and keep the sense of ownership and engagement by removing the option ‘D’. This way, students that preferred simpler instructions might have a better experience while the students that preferred creative instructions could still feel a sense of ownership.

4.2 Intervention Two

The second intervention examines the correlation between creativity preferences, self-efficacy, and student performance while students are taught new concepts. While the first intervention removed many extraneous variables, it excluded some of the data we desired to collect. For example, *Intervention One* examined the positive effects of creativity on student engagement while reviewing concepts. However, does teaching new concepts add complications that minimize the positive preferences? In addition, how does creativity impact other aspects, such as student efficacy and comprehension of material? The second intervention collects this data by gathering student performance prior to the treatments; teaching a new concept with different treatments (creative and non-creative worksheets); and comparing student opinions, performance, and efficacy against this intervention’s and the previous intervention’s results.

This intervention was originally designed as a three week experiment; however, stu-

dents took a week longer than expected to complete the tasks. The first week was to apply a baseline survey to assess their performance and efficacy on the previously learned subject (variables). No worksheet design was needed on our end for the variable lessons since this was taught by their teacher and was already in the curriculum. We also began teaching the new concept (conditionals), and the students began their worksheets. Week two, they were given a demonstration on conditionals before continuing the worksheets. Week three, they continued working on the worksheets. Week four, they finished the worksheets and received a closing survey.

4.2.1 Design

Worksheet Design

The conditional worksheets were modelled after the variable worksheets and other familiar worksheets. They were also vetted by the creators of the fifth grade curriculum.

There were four worksheet packets designed for this experiment - *Conditionals Part One* and *Conditionals Part Two* each in a creative choice and non-creative choice variant. Each treatment began with the same introduction to the concepts, per the lab's usual structure, presented via a 10-20 minute lecture at the beginning of the class. These worksheets included examples and fill-in-the blank lines of questioning similar to worksheets they had been exposed to. After the lecture, the worksheets had independent work sections for the students to continue working on their own. This section accounted for the majority of the students' time working. It is also where the creative treatment was applied.

Conditionals Part One

For *Conditionals Part One*, the students were introduced to less than and greater

than comparisons and the truth value of different statements. For the independent work sections, the students were asked to use the mouse cursor position to make different pictures appear on the screen based on if the mouse cursor was on the top or bottom half of the screen. Some still found the concepts of code blocks and variables challenging. Those who did not find it as challenging and who finished early were asked to provide additional edits to their picture (e.g., changing background color based on if the mouse cursor was on the left or right half of the screen).

The students with the creative worksheets could design any picture to appear based on the mouse cursor position (Figure 17). The students with the non-creative treatment were asked to draw circle puzzles (Figure 18). In designing these worksheets, we considered making the non-creative treatment more engaging than circle puzzles (e.g., making a picture of a mouse and a cat). Even though the assignment lacked creative choice, this did not mean the pictures had to be bland. Furthermore, disengaging pictures may bias student results in favor of the creative treatment for reasons unrelated to not being given choice. However, the circle puzzles were chosen as the picture for three reasons. First, a more engaging image may lead to some emotional response and the students' preference on the assignment may be correlated to their preference of the image itself (e.g., a student dislikes or likes mice and cats), making the data harder to dissect. Second, students may dislike it for the additional difficulty of a more complex and engaging image. Third, from previous experience working with the students, we found students often claim ownership of an accident in their code which creates an interesting change to the images they are working on. We wanted to prevent this accidental illusion of creativity. Therefore, we determined circle puzzles would be best.

Conditionals Part Two

For *Conditionals Part Two*, the students continued practicing conditional statements



Figure 17: *Conditionals Part One*: Example Picture Created by a Fifth Grade Student in the Creative Treatment Group.

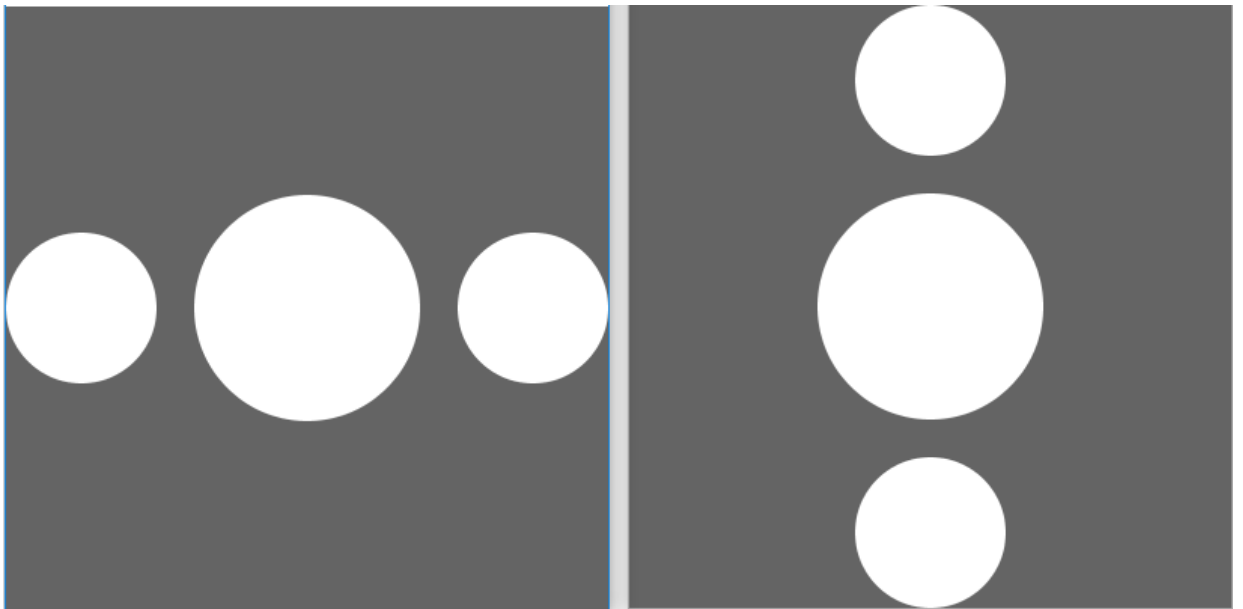
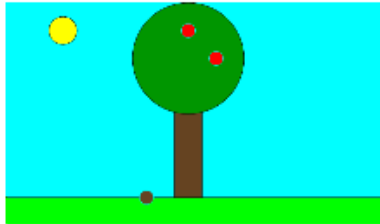

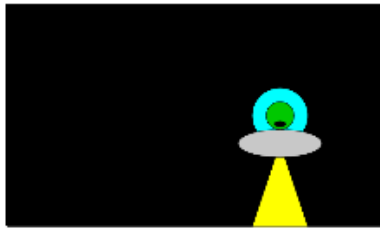


Figure 18: *Conditionals Part One*: The Two Circle Puzzles the Students in the Non-Creative Treatment Were Asked to Make.

with the mouse cursor, but were additionally introduced to using other variables (such as the position of a moving ellipse) in conditional statements. Students with

the creative treatment were asked to build any picture with moving pieces, applying different conditional statements to swap elements of the picture (e.g. Figure 19). Students were given multiple choice suggestions on images to make, but not told how to make any particular image. The suggestions were merely for giving students ideas. Students with the non-creative worksheet were asked to apply the same type of conditional statements to a moving ellipse (example step in Figure 20), but with explicit instructions and more restricted choice compared to the creative treatment. They were only allowed to apply changes to an ellipse as opposed to a picture of their choosing.

2. Add at least one conditional statement that changes something based on the moving piece's position

<p>A. When the position of the fruit hits the ground, have it stop moving. (What number will you need to use in the conditional for this?) Maybe you could make the apple shrink or change colors when it hits the ground as well!</p> 	<p>B. When the sun crosses to the bottom half of the screen, make the sky and grass darker</p> 
<p>C. If the UFO crosses to the right half of the screen, add a tractor beam</p> 	<p>D. Anything else you can think of!</p>

Go to the next page for the next step!

Figure 19: Conditionals Part Two Worksheet Segment: Example Challenge Assigned to Students in the Creative Choice Treatment.

Challenge 1

Fill in the code below to do the following:

1. If the ellipse is on the left half of the screen, make the size of the ellipse grow.
2. If the ellipse is on the right half of the screen, make the ellipse 50 wide and 50 tall.

Figure 20: Conditionals Part Two Worksheet Segment: Example challenge Assigned to Students in the Non-Creative Choice Treatment.

It is important to address the styles of creative choice embedded in each worksheet. In *Intervention One*, students were closely guided in their design choices through the multiple choice format of the worksheets. This is a suggestive and guided style of creativity. The *Conditionals Part One* worksheet differed as students were not given suggestions on what images to create. This resulted in students in the creative treatment spending much of their time brainstorming or making aesthetic design decisions. The *Conditionals Part Two* worksheet had multiple choice, similar to *Intervention One*. The multiple choice gave suggestions for the students on what to make (but unlike *Intervention One*, had no code support). Throughout our analysis, when the term ‘creativity’ is used we are referring to the either the open style or suggestive style of creativity, depending on the treatment being referenced.

The creative worksheets can be found in Appendix H and the non-creative worksheets in Appendix I.

Survey Design

There were a total of four surveys for this intervention. Two were given prior to the treatments to gather initial data on student efficacy, opinions, and performance scores on the material they were learning prior to our intervention (variables). Two were given after the treatments to gather the same data on student understanding of the new material (conditionals). The questions were distributed using google surveys, which were anonymous and matched to previous responses by student lunch IDs.

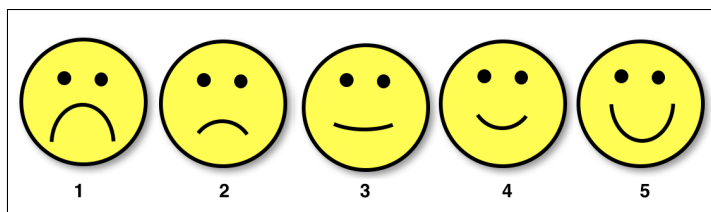


Figure 21: 5-point Likert Scale Used for Efficacy Surveys.

Students were assured we wanted their honest answers and that their responses would not reflect poorly upon them.

Survey One: Variable Self-Efficacy/Opinions

The first survey was a self-efficacy and opinion survey. First, we asked self-efficacy questions on student ability to use variables in Processing. While we tried to find previously validated computer science efficacy tests, the ones we found were either archaic, testing concepts beyond the scope of fifth graders, or for domain-specific languages that did not apply. One survey was well-written (though made for middle-school students), but it was published this year and not acquired in time for us to use [56, 68]. Therefore, we created domain-specific efficacy surveys, with statements modelled on Bandura’s Guide to domain-specific efficacy surveys [14]. The statements we asked students to examine were:

- I can write code to control the speed and direction of an ellipse using variables.
- I can figure out how to use variables to change the size of an ellipse.
- I can figure out how to use variables to change the position of an ellipse.
- I can figure out how to use variables to change the color of an ellipse.
- I understand variables and can help other students if they are confused about them.
- I can ask for help about variables if I don’t understand something in the lessons.

Although we began validating these questions, our surveys still need further validation to determine if they are accurately testing variable self-efficacy. In addition, contrary to Bandura's recommendation, we chose to use a 5-point Likert scale with pictures (see Figure 21) to simplify instructions for the students.

After the efficacy questions, we gathered data on student opinions about the quality and creativity of the worksheets. These questions mirrored the checkbox selections from *Intervention One*. Finally, we asked the students to select if they: wish they had more creativity on the worksheets, liked the worksheets as they were, or wish they had more rigid instructions. This was followed by a free-response explanation of their choice.

Survey Two: Variable Performance

The second survey was a performance quiz, testing student ability to answer questions about variables. The questions were split into general comprehension questions and application questions. We attempted to phrase the questions in a way the students could easily understand; however, they had never been given quizzes in computer lab before. Although we tried to explain their score was not a reflection on them, but on our teaching, they still showed some initial test anxiety which may have affected their performance.

Survey Three: Conditional Self-Efficacy/Opinions

After giving the two conditional treatments, we gave two more surveys to the students. Survey three was modelled on the *Variable Self-Efficacy/Opinion Survey*. We created and asked domain-specific efficacy questions on student ability to use conditionals. The statements we asked the students to examine are as follows:

- I can write conditional statements to check if the mouse is on the left half of the screen.

- I can change the picture that shows up on the screen based on a conditional statement being true or false.
- I can figure out how to write a conditional statement to check if a moving ellipse is on the left half of the screen.
- I can understand conditional statements enough to do the exercises we are given.
- I understand the difficult parts of conditional statements beyond just being able to do the worksheets.
- I understand conditional statements enough to help other students if they are confused about them

In addition, unlike *Survey One*, we also included self-regulatory efficacy questions developed in external research [31]. We wanted a validated way to measure self-efficacy outside of just using our newly developed surveys. *Survey Three* also included similar opinion questions about the quality and creativity of the conditional worksheets.

In choosing the self-regulatory efficacy survey, we researched numerous survey options. While many validated efficacy surveys exist (such as the MSLQ) for academic achievement/student learning, most are targeted for college students [56, 55]. Other studies have faced similar problems in finding age-appropriate computer efficacy surveys [11]. There also are some efficacy surveys refined for children, but some were phrased in ways that seemed susceptible to stereotype threat and others were validated against measures besides academic performance [41, 31]. Due to the limited options, but the desire to use at least one survey that underwent more extensive validation than we had resources for, we decided to use the following child self-regulated learning efficacy survey created by other researchers [31]:

- I can finish my homework assignments by deadlines.

- I can get myself to study when there are other interesting things to do.
- I can always concentrate on school subjects during class.
- I can take good notes during class instruction.
- I can plan my schoolwork for the day.
- I can organize my schoolwork.
- I can remember information presented in class and textbooks.
- I can arrange a place to study without distractions.
- I can get myself to do schoolwork.

The strengths of this survey were that it was designed for elementary students in the US (many surveys were for the wrong age group or phrased poorly for US students) and that it tested what we wanted to test. It also underwent inter-rater reliability tests and was shown to correlate with other aspects efficacy is expected to correlate with. One of its weaknesses is that it was not validated against academic achievement, which is an important step in validating student efficacy surveys [14]. Therefore, even though this survey was validated, we still needed to test its correlation with student performance.

Survey Four: Conditional Performance

This survey was modelled on the *Variable Performance Survey*, replacing questions to apply to the new topic of conditionals.

The four surveys for this intervention can be found in Appendix J.

4.2.2 Research Questions - Design, Results, and Threats to Validity

Research Question Concerning Baseline Performance

This section of our research reports baseline measurements of student performance in the different lab periods.

Research Question 7: Prior to applying the creative treatments, are differences in student performance, across the different lab sections, statistically negligible?

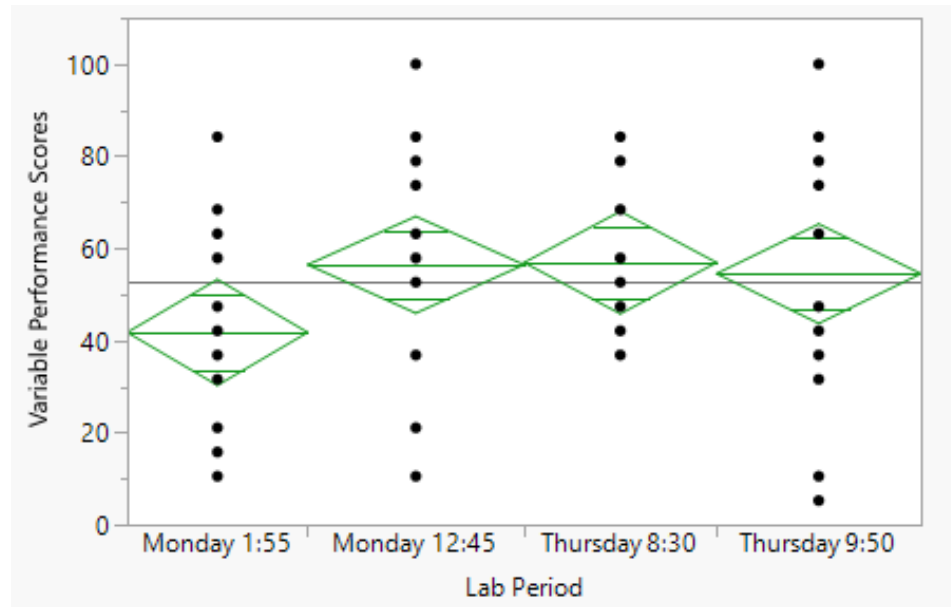


Figure 22: Oneway Analysis of Variable Performance Scores by Lab Period.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Lab Period	3	2385.611	795.204	1.6051	0.1973
Error	62	30716.463	495.427		
C. Total	65	33102.073			

Figure 23: Analysis of Variance on Variable Performance Scores by Lab Period.

Design

A goal of this intervention is to compare performance differences between classes that have learned material through a creative versus a non-creative treatment. Within education, however, there are often many confounding variables (e.g., different overall ability and learning paces across classes). When we were ready to implement *Intervention Two*, the students were finishing lessons on variables. We applied the *Variable Performance Survey* to evaluate if the classes were performing similarly prior to our intervention; if they performed the same, we could meaningfully attribute performance differences after our intervention to our treatment. In addition, this was the first time the students had been quizzed on the computer science lab material, lending another important reason we provided this baseline survey experiment.

Results

Our hypothesis was that the different classes would perform similarly on the *Variable Survey*. After running ANOVA to find the variable performance score variance by lab period, we found a p-value of 0.1973 (see Figure 23). This is not significant, indicating that the scores are not statistically different. However, the Monday-1:55 class does have a lower mean score than the other classes (Figure 22). Although this is not significantly lower, we still discuss solutions to the score differences in the following section.

Threats to Validity

Although there were no statistically significant differences between the mean scores of the classes, it is clear that the later Monday class did not perform as well as the others (Figure 22). Therefore, in later analysis, while we provide statistics on all of the lab periods' data, we also provide statistics on just the Thursday labs; these classes had closer performance means and were taught on the same day, removing some extraneous variables. Although this should not be necessary based purely on

the p-value found, it is an extra precaution we take to address this threat moving forward.

Research Question Concerning Material Validation

The following research question addresses validation of the three efficacy surveys.

Research Question 8: Do our three efficacy surveys (on self-regulated learning, variables, and conditionals) correlate with student performance?

Design

According to Bandura's guide to creating efficacy surveys, efficacy results should correlate with academic performance [14]. To validate our surveys, we analyzed the correlation between the performance scores with the self-regulated efficacy scores, the variable performance scores with the variable self-efficacy scores, and the conditional performance scores with the conditional self-efficacy scores.

Results

The self-regulated efficacy scores do not correlate with either the variable or conditional performance scores. The Pearson Correlation Coefficients are 0.1206 and 0.1806 respectively, which indicate no significance. We also binned efficacy scores into high scores (4-5) and low scores (1-3) to run ANOVA on performance by high or low efficacy; we were unable to reject the null hypothesis that the mean performance scores were the same (as seen in Figure 24). There should be consistency between efficacy and academic performance, suggesting that either the survey or our performance quizzes were unreliable.

Variable self-efficacy also does not correlate with variable performance scores. The

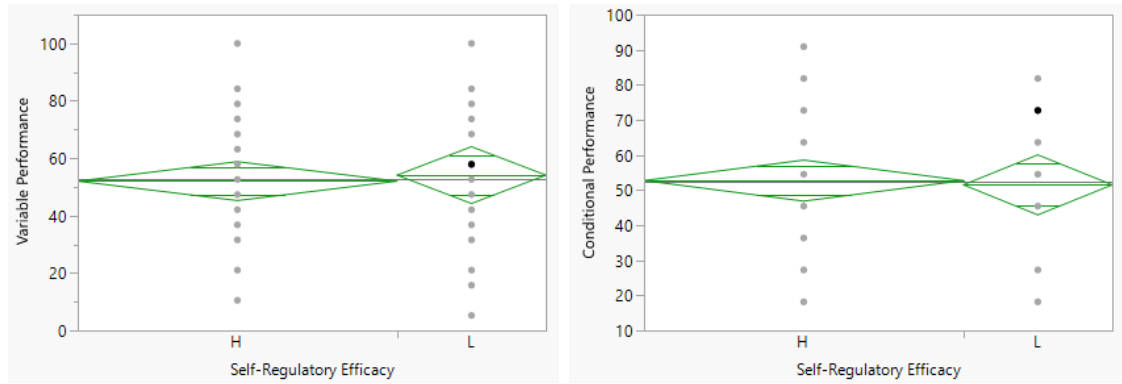


Figure 24: Oneway Analysis of Variable and Conditional Performance Scores by Self-Regulatory Efficacy rankings of High and Low.

Pearson Correlation Coefficient is 0.2294 with a p-value of 0.0639 - this is not statistically significant; running ANOVA also does not show promising results, as seen in Figure 25.

However, conditional self-efficacy *does* slightly correlate with conditional performance scores. The Pearson Correlation Coefficient is 0.4482 with p-val .00016; this is, though not a strong percent correlation, statistically significant. In addition, when binning efficacy scores into high and low efficacy and running ANOVA, we are able to reject the null hypothesis (with a p-value of 0.0014) that the scores were the same for students with high and low efficacy scores. This supports that the mean performance scores of students with high self-efficacy are statistically higher than performance scores of students with low self-efficacy (Figure 25).

Overall, our conditional self-efficacy survey paired with the conditional performance survey was the most reliable of the three. The correlation indicates that both conditional surveys are valuable measurement tools. By doing initial testing with the variable surveys, we were able to refine our processes and familiarize the students with the styles of surveys they were given to create useful measurement tools for the remainder of our experiment.

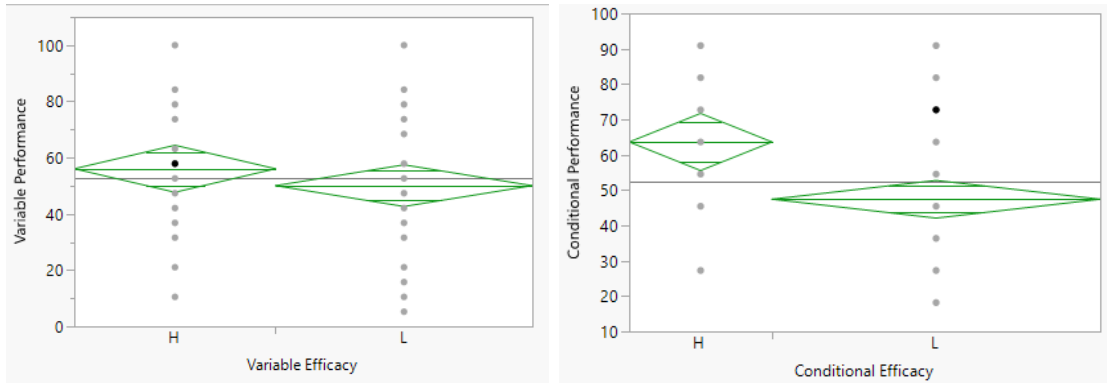


Figure 25: Oneway Analysis of Variable and Conditional Performance Scores by Variable and Conditional Self-Efficacy Scores.

Threats to Validity

The self-regulated efficacy survey was previously validated against student psychological measures, but never against student academic achievement. We attempted to further the work of validating it by comparing efficacy scores to student academic achievement. The lack of correlation *could* indicate an incorrect experiment; we used our performance surveys to measure academic achievement, but it is likely these quizzes are not an accurate measure of student academic achievement in their fifth grade class. In the future, this efficacy survey should be validated against student grades, not student performance on computer lab surveys. Another cause of the uncorrelated data could be faulty performance surveys. Perhaps our surveys do not accurately measure student understanding on the material; however, due to the positive correlation found for our conditional surveys, we have some indication of accurate performance measurements. Alternatively, the cause could be that the self-regulated efficacy survey is not an accurate measure of self-regulated efficacy. It is uncertain what caused our results. The domain-specific variable survey also faced similar threats.

However, we *did* find positive (though weak) correlation for the conditional survey. It is possible that the uncorrelated results for the variable surveys was because this

was the first time giving this style survey to the students; both the students and the instructors learned from this first application of surveys. We learned how to better explain the efficacy tests after administering the variable surveys *and* the students gained more self-awareness; many students reported high variable self-efficacy and then saw how poorly they did on the variable performance quiz. This could have helped their awareness on conditionals.

Overall, this section reveals some potential flaws in the measurement materials. However, while the variable surveys were not promising, they helped refine the process for distributing our conditional quizzes (which were more important to our study). In addition, though we did not find promising results concerning the self-regulatory efficacy test, we can still cautiously use it for analysis, seeing as it was validated in other research.

Research Questions on Performance and Efficacy by Treatment

This section experimentally analyzes how our creative and non-creative treatments affect student performance and efficacy.

Research Question 9: Does learning new material with different treatments (creative versus non-creative) affect student performance?

Research Question 10: Does learning new material with different treatments (creative versus non-creative) affect students' domain-specific self-efficacy?

Design

The creative and non-creative worksheets, conditional performance survey, and conditional self-efficacy survey designs are discussed in the *Worksheet Design* section. 53 students learned about conditionals with the creative set of worksheets and 54

students learned about conditionals with the non-creative set of worksheets. After, the students were given the conditional performance and conditional efficacy surveys.

Results

Our hypothesis was that the students in the creative treatment would have higher self-efficacy due to increased engagement while maintaining similar performance scores. However, ANOVA for conditional scores grouped by lab periods and ANOVA for conditional scores by treatment indicated no statistically significant difference (Figures 26, 27, and 28).

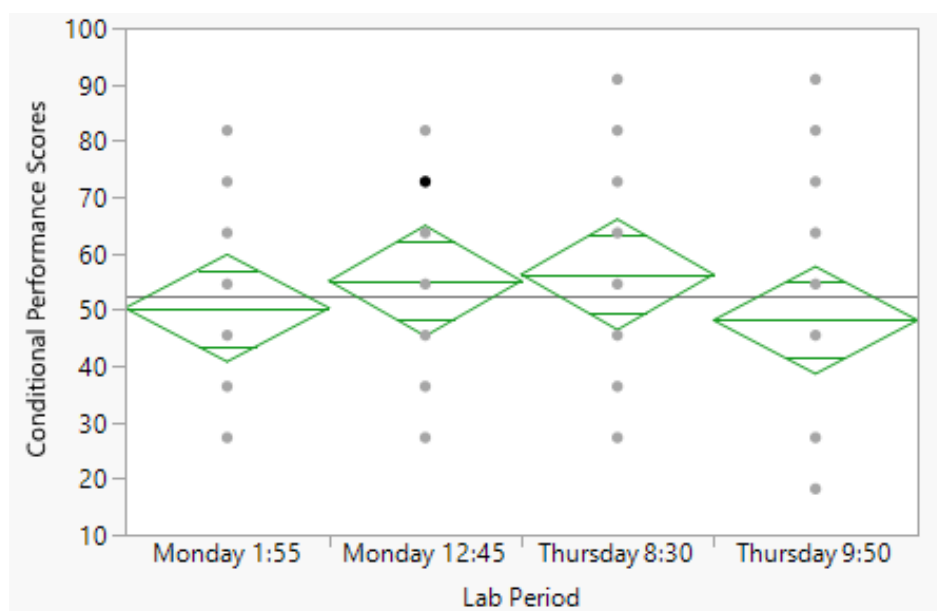


Figure 26: Oneway Analysis of Conditional Performance Scores by Lab Period (All Periods - Monday-1:55 and Thursday-8:30 Labs Were Given the Creative Treatment)

We also analyzed the statistics of the Thursday classes, as discussed in *research question 7*; their performance differences also were not statistically significant (Figure 29).

Running ANOVA on conditional self-efficacy scores by treatment also indicated that any difference between scores was not significant (Figure 30). Overall, our treatments

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Lab Period	3	742.273	247.424	0.6420	0.5909
Error	62	23895.843	385.417		
C. Total	65	24638.117			

Figure 27: Analysis of Variance on Variable Conditional Scores by Lab Period (All Periods - Monday-1:55 and Thursday-8:30 Labs Were Given the Creative Treatment)

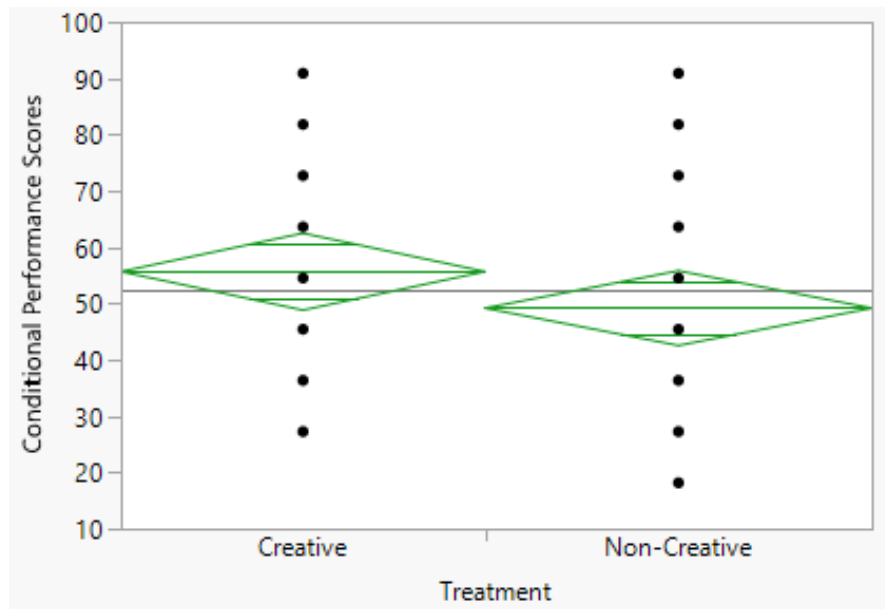


Figure 28: Analysis of Variance on Variable Conditional Scores by Treatment (Creative and Non-Creative) (P-Value of 0.1783 Shows the Mean Difference is Not Statistically Significance)

did not seem to affect performance or self-efficacy.

Threats to Validity

One threat is that education research almost always has more variables than intended. For example, we designed the two sections of the worksheets to be as identical as possible aside from levels of creativity. However, in attempting to design those two treatments, we could have introduced other unintended variables - for example, some students found the non-creative instructions to be confusing, even though they were

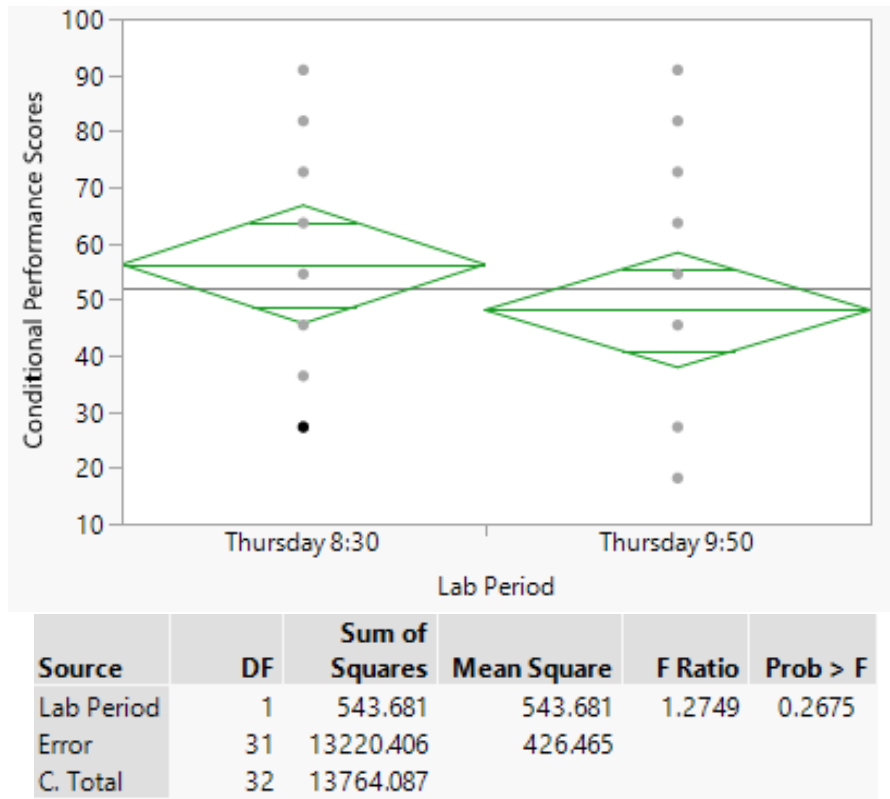


Figure 29: Oneway Analysis of Conditional Performance Scores by Thursday Lab Period (8:30 Lab Receiving Creative Treatment and 9:50 Receiving Non-Creative).

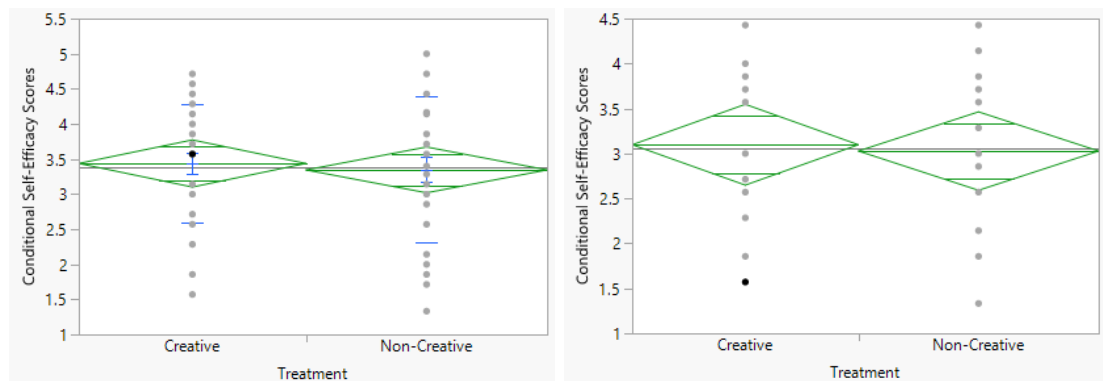


Figure 30: Oneway Analysis of Conditional Self-Efficacy Scores by Treatments (Creative and Non-Creative) for All Classes (Left Image) and for Just the Thursday Classes (Right Image). (The P-Values of 0.6970 (Left) and 0.8206 (Right) Indicate No Significant Difference in Efficacy by Treatment)

intended to be straightforward. It is hard to know if these select students would have been confused regardless, or if we unintentionally created difficult instructions.

In addition, this was a very short treatment, making it difficult to cause large learning/efficacy differences.

Research Questions on Creativity Opinions by Treatment

The worksheet design, as discussed in the *Worksheet Design* section, consists of two part worksheets in both a creative and non-creative variant.

Research Question 11: What are student opinions on the amount of creativity given in the variable worksheets and the different conditional treatments? Do responses make sense given the treatment?

Design

To test student opinions across *Intervention Two* worksheets, we asked student to select if ‘I wish we had more creativity or freedom in these worksheets’, ‘The worksheets had a good amount of creativity and instructions’, or ‘I wish we had more rigid instructions’. The students were then asked to explain why they made their selection.

Results

The results for student preferences did not seem to match the treatment they were given. Table 1 shows that students in the non-creative treatment actually wished for more rigidity after being given the non-creative worksheets. This could be due to the fact that some students found the instructions confusing. Just because the non-creative worksheets lacked creativity does not mean they were concise and clear. In the future, we would like to refine our non-creative conditional worksheets to make them more comprehensive for students and run the experiment again.

As predicted, most students who wanted rigid instructions reported that the instructions were confusing or that Processing is hard. A few students had slightly different reasons. One student reported they wished for rigid instructions ‘Because I want to make more interesting pieces by following instructions. Another reason why I want more rigid instructions is because it can help me remember what to do in order to create something else for a certain code.’ And another shared: ‘I sometimes cant think what to type.’.

Variable Responses		
Wished for Creativity	Good As Is	Wished for Rigidity
17	14	3
Conditional Responses		
Wished for Creativity	Good As Is	Wished for Rigidity
16	8	10

Table 1: Count of Student Preferences in No CC Treatment (Note: Students All Received the Same Treatment for the Variable Sheets).

Table 2 shows the student responses before and after receiving the creative treatment. There was not much change, possibly indicating that this version of creativity did not satisfy the students’ desires for creativity, nor did it seem to match their definition of creativity.

Variable Responses		
Wished for Creativity	Good As Is	Wished for Rigidity
13	14	5
Conditional Responses		
Wished for Creativity	Good As Is	Wished for Rigidity
14	13	5

Table 2: Count of Student Preferences in CC Treatment (Note: Students

All Received the Same Treatment for the Variable Sheets).

However, when looking at some of the student free-response explanations, some students seemed to have a more positive perception than the numbers imply. Some responses of students who still wished for more creativity after were as follows:

- *“It’s already creative BUT i want it to be more creative because <https://valentin.dasdeck.com/processing/index.php> is very fun to use and i want it to be more creative because it might help with in the fucher”*
- *“Well, there was much creativity and freedom in these worksheets, but I feel there should be more because they told us what to do mostly. I also think that we should have a paper that we make what we want.”*

Therefore, these results, instead of indicating flaws in the creative worksheet, further indicate the importance of creativity for student engagement. Even after being given a creative assignment, students still wanted more.

Threats to Validity

Student responses, especially for the non-creative treatment, did not change as expected - more students reported a desire for rigid instructions after receiving the non-creative assignment than they did before. This leads us to believe that our worksheets were not concise enough or that students did not understand the intent and phrasing of the surveys.

4.2.3 Summary

In this intervention, we established that prior to our treatment, the four lab periods performed similarly. We also began validation of our conditional self-efficacy survey.

Next, we found that creative and non-creative treatments do not impact domain-specific self-efficacy nor performance. Finally, we analyzed student opinions on the worksheets they were working on immediately prior to our intervention and opinions on the creative and non-creative worksheets about conditionals.

Chapter 5

JUVENILE HALL CURRICULUM

This section of the thesis reports on the design and implementation of a computer science class at San Luis Obispo County’s Juvenile Hall.

Research with the fifth grade curriculum was motivated by the finding that students reported a strong enjoyment of art. We explored how creative choice engages students within the context of fifth grade and the context of computational art. Since choice seemed to be a big component of engagement with the fifth graders, we wanted to explore how game design, a topic embedded with even more choice, could engage this older group of students.

5.1 Design

Overall, the goal of this curriculum is to give opportunity to an underserved population, specifically that of San Luis Obispo’s Juvenile Hall students. Computer science is a growing and lucrative career path, making it unjust that currently only some populations have access to it. However, there are more considerations to take into account than merely distributing curriculum to everyone; the curriculum must be designed carefully or it may do more harm than good [44, 51]. Therefore, it is important to consider *what* type of course it is (an introduction to computer science) and *who* the course is for (high school, Juvenile Hall students).

5.1.1 What: Introductory Course

As an introductory course, the goal of this class is centered around retention and inclusion over technical depth. As an independent elective for high school students, there are few repercussions if students do not fully progress through the material; conversely, there are serious repercussions if students are deterred from the major, foster stereotypes, or do not feel empowered. Therefore, the design of the class focuses on the latter.

5.1.2 Who: Demographics

This curriculum was planned and tailored to serve the target population.

San Luis Obispo County’s Juvenile Hall and CVA Youth

San Luis Obispo County’s Juvenile Hall is divided into two separate onsite facilities. The main detention facility, Juvenile Hall, is a “50 bed detention facility for male and female youth who have been arrested for criminal acts in the community or violations of probation” [8]. The second program is Coastal Valley Academy (CVA), a “custody commitment camp program in Juvenile Hall for 14-17 year old male and female youth who are moderate to high risk and in need of residential treatment” [8]. It is onsite at Juvenile Hall, but has separate programming, living units, and a classroom. Youth stay here 6-12 months, which is much longer than most youth stay at Juvenile Hall. CVA youth also receive “intensive case management, treatment and educational services” during their residency, as “staff are trained in Comprehensive and Substance Abuse Curriculums and Positive Behavioural Interventions and Supports (PBIS)” [8].

Restorative Partners offers numerous programs at Juvenile Hall. When first planning with Restorative Partner’s Volunteer Programs Director, Robyn Morton, she deter-

mined the CS course should be taught at CVA. Since detainment is 6-12 months, *most* student would be able to stay for the duration of the four to five week course; however, we also knew students would be dropping and adding throughout due to the nature of CVA.

The Students

The curriculum for CVA youth is designed for middle or high school students. Students are “moderate to high risk and in need of residential treatment” [8]. Therefore, a curricular focus was on making sure the class was empowering and fun. The curriculum was designed to challenge them, but include immediate encouragement and reward as well.

However, there were also many unknowns during the planning stage. We were uncertain of the size of the class and the exact grade of the students. In addition, due to the flexible nature of CVA and the uncertainty of when students would join or leave, the course had to be rescheduled multiple times. Due to many unknowns working in this environment, we incorporated flexibility into the curriculum to allow for adaptation to the needs and interests of the students as they arose; we also always had extra activities planned in case we needed to change course. Finally, we had base code, worksheets, and activities prepared for students who joined late in the course.

While planning, it was also uncertain if students had previous experience working with computers. We designed the curriculum assuming they had basic typing skills, but were also prepared to adapt as needed.

5.1.3 How: the Curriculum

Research Considerations

As mentioned in the Chapter 2, to design the curriculum for an introductory CS

course, we utilized research on college CS courses for non-majors and research on teaching high school students CS. Research found that project-based learning is engaging for non-majors and can increase self-efficacy [29, 46]. In addition, there is a need for both breadth and depth in an introductory CS curriculum [58, 64]. Finally, research shows the benefits of proximal subgoals and the idea of immediate rewards [16]. We attempted to incorporate these findings through a 2D game design course.

Research on teaching high school students shows many young adults have stereotypes that CS is anti-social and boring; we utilized the researchers' suggestions for correcting those misconceptions by using game elements, fun projects, real-world applications, and creativity [74].

Curricular Design

Overall, the curriculum was built around creating a simple 2D game. This way, the youth would have a project to work on which they could hopefully be proud of and feel ownership of, as recommended by researchers mentioned above. Throughout the five weeks, each class consisted of a 10-20 minute lesson to teach a new coding concept and then an assignment or two to practice the new concept. Each assignment would directly add to the students' games. This way, they would practice the lessons of that day while still working towards an immediate subgoal for their own project.

We chose to use Processing as the language so we could adapt many of the same ideas from the fifth grade curriculum. In addition, Processing is text-based and simple, providing immediate visual feedback for the students. We considered making a computational art class instead of a game design class to more closely follow the fifth grade curriculum. In addition, this would be a simpler introduction to coding than creating a game, since games introduce more complex math concepts. However, we decided creating a game would be more engaging and worth the added difficulties.

We asked the youth their opinions on those two options in the post-survey after the class was over.

Overall, although we did diverge from the fifth grade curriculum, we still were intentional about weaving creativity throughout the lessons. The students were allowed to design their game characters given the constraints of the commands they knew and design or choose all of the elements in their game. In addition, although we had instructions and outlines for their game, as much as possible we tried to accommodate the student requests to customize different elements of play.

The syllabus for this course is found in Appendix K.

Constraints

While designing the curriculum, we had many constraints to consider. First, many CS0 classes have lectures, lab assignments, homework, and projects. Coding is a difficult skill to learn, and often requires multiple facets to practice. However, this class only had two one-hour periods each week. In two hours a week, we needed to provide all of the lectures and assignments, projects, and work periods. The students were unable to work on the course material outside of the hour blocks, partly due to their schedules and partly because we brought in our own laptops each week. (Bringing in laptops that could not connect to the internet made the process much smoother. Also, when originally planning the course we did not know if there would be other computers available to the students.) Without being able to assign homework, it was essential to have every assignment work towards the overall game the students were making. We hypothesize this was more engaging, working towards a goal instead of doing isolated practice problems; it was necessary to make time for all of the lectures, labs, and project work in an hour period.

Another way we accommodated for the time constraints was by having two other

Cal Poly students (Timothy Wong and Erik Mork) come in each week to assistant teach. Theresa Migler, who had experience teaching at San Luis Obispo County's Jail, emphasized the importance of having extra volunteers; we found the class would have been impossible to teach in the style we did without TAs. Having two volunteers ended up providing a one-to-two ratio of TAs to students during most of the course. While the students stayed engaged during the lecture and learned overall concepts during that time, 10-20 minutes was not long enough to make much learning progress. Therefore, the tutoring-style work period allowed them to get the help needed to make progress on their projects and aided in reiterating the lessons.

Finally, the last way we accommodated for the time constraints was providing scaffolding for different parts of the project. In addition, while they designed their project and coded the first part of each assignment every class, one hour was never long enough to learn and then finish an assignment on a new topic. After each class period, with the students' permission, we finished the remainder of some of the class assignments so they would be ready to move on to the next lesson by the next class. The decision to move along through the curriculum this way was not originally in the design, but emerged after the first few classes and after talking with the students' regular teacher. He explained it would be most encouraging for the students if, once they started learning a new concept, they were not bogged down by tedious tasks when time was so limited and holding their engagement of utmost importance.

5.2 Implementation

Creating and launching this course consisted of two phases: setup and teaching.

5.2.1 Setup

Before the class could be taught, volunteers needed to be cleared to enter Juvenile Hall and trained on rules and regulations. Robyn Morton, Restorative Partner’s Volunteer Programs Director, aided in these steps. Another setup requirement was determining how to make this course count for the students’ high school credits, as each student had different course requirements remaining. Some youth were able to count this as an general elective credit; however others needed it to be a math elective or art course instead. The range of course needs of these students supports Bootstrap’s idea of incorporating multiple subjects while teaching CS curriculum. Since our course was a combination of 2D game design and computer science, it was heavy in math, coding, and art/design; this enabled it to count for diverse course needs.

5.2.2 Lessons

The original curriculum consisted of eight lessons (two per week), though this was later extended by two lessons to enable the students to work at a more appropriate pace. Overall, there were ten classes over the course of five weeks. The course outline is included below.

Lesson 1 - Introduction to Computer Science

Class Overview:

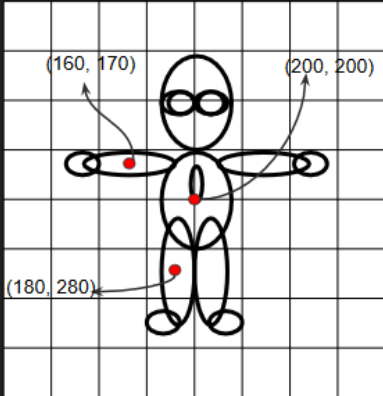
This class consisted of the pre-survey (found in Appendix M), followed by a Power-Point aided lecture on general computer science topics. The presentation covered the goal of the class as a 2D game design course and how it fit into the context of CS as a whole. It also examined the various types of computer science as well as the many fields and applications it extends into. Next, we covered a very high-level overview of software versus hardware, code, programming, and programming languages. We

discussed 2D space, interactively going over graphing points in a ‘normal’ grid versus screen-space. (Students were asked to come up to the board to plot points in the two different spaces). Finally, we outlined the 2D game they would make (a simple game where a character slides across the screen to collect falling objects).

Assignment:

The first assignment, for the last 10 minutes of class, was to design the main character of their game. The constraints and requirements can be seen in Figure 31. The students were given a grid to draw on and to label their shapes.

First Assignment:



- Draw your hero
- BE SURE TO:
 - THINK SIMPLE (5-20 shapes)
 - Only use simple shapes (triangles, circles, squares). Label the points for the:
 - Center of the circle
 - Each corner of a triangle
 - Top left corner of the square
 - OR make shapes by connecting the dots with straight lines between points
 - Label each point
 - Only connect dots with straight lines
- Following these steps will help you when you turn your drawing into computer code next class!!!

Figure 31: First Assignment for Students to Start Planning Their Game.

Lesson 2-Processing Commands to Draw Hero

Class Overview:

Class began with a PowerPoint outlining basic Processing commands to enable the students to draw the character they designed on the computer. The presentation included a discussion on how selecting the screen size is like picking a canvas, how typing the rgb color command is like selecting paint for a paintbrush, and how the different shape commands is similar to painting. We explained the rgb color wheel, reviewed how to call the basic commands, and discussed how order of commands

matters. This also was compared to painting: just as you must pick a color before painting, Processing developers have to write the color command before the draw shape commands. Furthermore, just as painting new shapes over previously painted shapes covers them, the commands for coding a picture in Processing also layer most recent commands on top of older ones. Students were given a ‘cheat sheet’ to assist them while they coded since their laptops did not connect to the internet.

Assignment:

The assignment was to write the code to draw their hero. (Note: to keep the students on track, we finished all but the last few lines of code for each student after class. Overall, we did about a third to two thirds of the character - it varied by students.)

Lesson 3-Functions

Prep Work:

The students were unable to finish their character code during lesson two, though all had made good progress. To move the lesson along and keep the students engaged, we finished writing all but a few commands on their characters prior to lesson three. In addition, students had not yet learned about functions, so their code from lesson two was written linearly. When assisting with their code, we also organized their code into functions, a *setup* function and a *draw* function. In Processing, these are recognized functions used to enable 2D animation. *Setup* is called once per program run and *draw* is a render loop called each frame. By arranging their code for them, we were able to efficiently introduce functions at the beginning of lesson three, as well as to start teaching the students how to read code that was not written by them. In addition to prepping the students’ code, we also wrote a separate program with an empty *draw* function and three different functions to draw three different background scenes. The sample backgrounds can be seen in Figure 32.

Class Overview:

The lesson was on functions, explaining the built in *setup* and *draw* functions, as well

as functions they could make for other purposes.

Assignment:

Students finished their characters, becoming familiar with the new code organization and reminding themselves of the Processing commands. After, students were instructed to apply what they learned about functions to call the different backgrounds in the background program we had made. After familiarizing themselves with the background program, they copied the function for their favorite background into their game code and called it in the *draw* loop. This allowed them to practice using functions, reading code, and integrating preexisting code into their program. It also saved a lot of time by not forcing them to create an entire background from scratch.

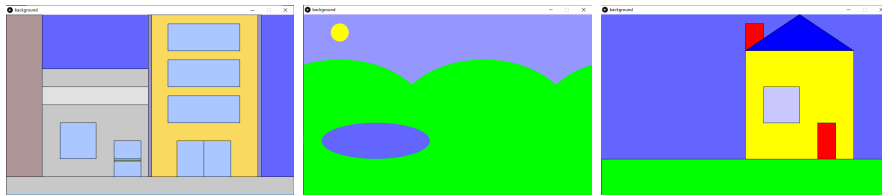


Figure 32: Backgrounds Students were Given to Chose from While Practicing Function Calls.

Lesson 4-Continuing Functions and Introducing to Variables

Class Overview:

Class began with a review of functions and with a simple practice exercise. Students were asked to move the code to draw their character from the *draw* function to a new function with a name of their choosing. After, students were taught how to translate their character to the bottom of the screen. Once that was completed, students were introduced to the topic of variables, with a white-board lesson; this was followed by a demonstration in Processing, which showed how a variable representing position can be changed to make shapes move across the screen.

Assignment:

Students were asked to move their character code to a function and translate the character to the bottom left of the screen. For the last part of class, students designed the falling objects their characters were supposed to collect since there was not time to add the code to make their characters move.

Lesson 5-Continuing Variables

Class Overview:

This class began with a white-board lesson on variable scope, where we discussed parameters, local variables, and global variables. We also discussed integers and floats and then outlined how to do the assignment.

Assignment:

The students were asked to use a variable representing position to slide their character across the bottom of the screen.

Lesson 6-Conditionals and Booleans

Class Overview:

This class introduced conditionals and booleans with a white-board lecture. We also demonstrated the use of conditionals in Processing and introduced pseudocode indicating how to do the assignment.

Assignment:

The students were asked to use conditionals to make their character stop moving when they reached the right side of the screen. Next, they used booleans to make the character move back and forth along the bottom of the screen.

Lesson 7-IO Key Controls and More Fun with Variables

Prep Work:

Since the students already had a good understanding of the basic draw commands, we used their ‘falling object’ design from lesson four to write for each student a function that would draw an instance of their falling object. This was necessary due to the

time constraints discussed previously. We also added most of the code for recognizing key presses, but left some lines for them to finish so they could practice working with this code.

Class Overview:

Students were given a quick lesson on IO and then given an explanation and outline of their next assignment.

Assignment:

The students were instructed to complete the KeyPress code to allow them to control their character's movement. They also were given an assignment to review variables, since the topic was still confusing for some of them. They practiced by creating different position variables for each instance of a falling objects they wanted. Next, they used these variables to draw and animate as many falling objects as they wanted on their screen at a time (most choosing to make three).

Lesson 8-Game Play

Class Overview:

This lesson was created to address different student requests throughout the course. Originally, students were all instructed to make a game with a main character and falling objects for the hero to collect. If the hero missed an object, they would lose. It was originally expected that all students would make the same style game so we could plan when to introduce the topics needed for each stage in the game's creation. Some students, however, indicated interest in creating a dodging game instead, where the character had to avoid falling objects and would lose if they were hit by one. To accommodate the student requests, we made lesson eight an outline of game play options. Students were given different assignments based on the game variants they chose. We gave an overview on how to make a game-over screen when an object hit the ground. This assignment required the use of booleans, basic draw functions, and using conditionals to check when objects reached the edge of the screen. These were

all topics students had previously learned. This assignment was given to students making a collection game.

We also gave an overview on how to make the falling objects cycle back to the top of the screen when they hit the ground. This assignment continued using variables and using conditionals to check when objects reached the edge of the screen. This was assigned to students making a dodging game. These different assignments accommodated student requests.

Assignment:

Students implemented the next step that applied to their specific game.

Lesson 9-Collision Detection

Class Overview:

The lesson was on how to represent collisions in a game using bounding box approximations. This was demonstrated with pseudocode and drawings on the whiteboard.

Assignment:

Students began implementing collision detection.

Lesson 10-Final Touches

Prep Work:

We finished the students collision detection so they would all have functioning games. In addition, this allowed them to focus on the fun, final touches of game creation if they wanted.

Class Overview:

Students were given a free work period.

Assignment:

Students could add last minute edits to their games, working on the design or gameplay.

5.3 Validation

This section discusses the research approval, surveys, and both the quantitative and qualitative results of the course.

5.3.1 Research Approval

As with the fifth grade curriculum, this research section also received IRB approval; it was especially important since we were working with incarcerated youth. The proposal and guardian permission forms are in Appendix B.

5.3.2 Surveys

Surveys were given both before and after the class to gather student efficacy and opinions. The surveys were anonymous and were given in the form of printed out Google form surveys (for the pre-survey, found in Appendix M) and questions on the white board (for the post-survey).

5.3.3 Quantitative Results

Research Question 1: Was game design an effective and engaging way to introduce our students to CS?

Our hypothesis when designing this curriculum was that game design, over computational art, would be the most engaging way to teach this group. Choice was found engaging for the fifth grade students, and game design holds even more decisions

than art. However, our pre-survey results showed that students had the exact same interest in learning to program as they did in learning to make games. This implies that other areas may have been equally interesting to these students.

However, although pre-surveys indicate games are no more enticing than computer programming in general, post-surveys show that most students would recommend a game over computational art for future iterations of this class (Table 3). Although this is not statistically sound data (the sample size is too small, and the students were biased by having just taken the game design course without taking an alternative course for comparison), it was still helpful in considering the class’s next iteration.

Game	Art	Either
4	0	2

Table 3: Number of Students (of Six) Preferring a Game-Design Course, Computational Art Course, or Either.

Research Question 2: What were student opinions concerning CS, games, coding, and creativity prior to taking this course?

From the pre-survey, students reported on if they would consider a career in computer science and if they thought they would be able to get a job in computer science one day if they wanted to. Students showed mixed results, which can be found in Table 4. They were also asked if they would like to learn computer programming and if they would like to learn how to make games on the computer. Students all had neutral or positive responses to these statements, which are displayed in Table 5. Finally, students were asked if they thought they were creative and if they liked when their classes let them be creative. Students, once again, had neutral to positive responses, which are shown in Table 6. Averages of student responses are displayed in Table 7.

This further supports the idea of using creativity to engage a broader range of students in CS.

I would consider a career in computer science	I will be able to get a job in computer science one day if I want to
6	6
6	5
3	2
4	5

Table 4: Pre-Survey Concerning Computer Science as a Career (7-Point Likert Scale)

I would like to learn computer programming	I would like to learn how to make games on a computer
7	7
4	4
5	5
6	6

Table 5: Pre-Survey Concerning Desire to Learn Programming Topics (7-Point Likert Scale)

I am a creative person	I like it when my classes let me be creative
7	7
4	4
6	6
6	6

Table 6: Pre-Survey Concerning Creativity (7-Point Likert Scale)

Averages	
I would consider a career in computer science	4.75
I will be able to get a job in computer science one day if I want to	4.50
I would like to learn computer programming	5.50
I would like to learn how to make games on a computer	5.50
I am a creative person	5.75
I like it when my classes let me be creative	5.75

Table 7: Averages of Student Responses from Selected Pre-Survey Questions (7-Point Likert Scale)

Research Question 3: After the course, did students report liking the class, a desire to continue programming, or a desire to pursue a career in computer science?

The results for the post-survey concerning these questions is found in table 8. The students were asked to answer questions on a 7-point Likert scale, with 7 being ‘fully agree’. However, since the post survey was hand-written, students were not forced to follow the directions as they would have been online. Note: for the question on if students liked the class, though the average score was 7, not all students gave the answer 7. One student had reported a 6 and another student reported liking the class 8 out of 7.

Averages	
I would consider a career in computer science	4.67
I would like to continue learning computer programming	6.67
I liked the class	7

Table 8: Averages of the Post-Survey Results for the Six Students Present

on the Last Day of Class (7-Point Likert Scale)

Research Question 4: Did student interest in CS increase after completing the course?

Although we did give pre and post surveys, only three students ended up being able to take both. The results for those students can be seen in table 9.

Pre-Course Surveys	Post-Course Surveys
I would consider a career in computer science	I would consider a career in computer science
6	7
4	7
6	7
I would like to learn computer programming	I would like to continue learning computer programming
7	7
6	7
4	7

Table 9: Comparison of Pre and Post Surveys for the Students Who Took Both (Other Students Joined Late or Left Early) (7-Point Likert Scale)

5.3.4 Qualitative Results

Overall, everyone involved - the CVA teacher, the students, and the volunteers - reported the class being a positive experience. The teacher was excited to have his students interacting with college students and hopes to continue this course in the future, as well as have other students from other majors teach. His report on the class can be found in Appendix L.

Positive Student Engagement

The students all seemed to enjoy the class, as we had positive experiences with every youth despite their initial levels of interest in the subject. For example, one student showed very little interest at first and wondered why he had to take the class - he already had chosen a career path outside of computer science. However, by the end of the course, although he still did not want to pursue computer science as a career, he was one of the most engaged students, working until the very end of class on the last day. He took initiative with his work, researching new commands and coming up with out-of-the-box ideas we had not taught. He even learned how to add lives and a high score counter on the last day, although this was outside of the scope of the class lectures. Overall, even though he does not wish to pursue this career path, he reported he would like to continue learning computer programming.

In addition, all students showed signs of having pride and excitement in their work. The last day, we gave students the option to demo their game to the class. While the students declined to demo in front of the class, every student, during the last class period and also throughout the previous classes, demonstrated their games to each other in one-on-one settings. The students would regularly and excitedly call the teacher, the volunteers, the guards, and each other over to their tables to show off new elements they had added to their games.

Benefits from Course Flexibility and Tutoring Style Work Periods

We found flexible curriculum and tutoring-centered work periods helpful in this class setting. Flexible game goals and having many opportunities for one-one-one help allowed the students to focus on the areas of the game they were most interested in, while less interested students still had basic, functioning games by the end of the course. One student, for example, was most interested in the artistic and design aspects of making a game. Although he did not pick up complex coding concepts as

fast as some students, he worked very hard and stayed focused on his work. He was the only student who asked if he could change the default backgrounds given to him and spent much of his time perfecting it.

In addition, we still found the class to be a positive experience for students joining late. In this setting, it is very common for attendance to fluctuate. Students who joined the course halfway through were given base code and able to integrate into the class. They were given extra help from tutors and other students and were able to complete a game by the end. However, one student showed up so near the end, we were unable to teach him how to make a functioning game. For this student, we taught him the basic draw commands and had him make static drawings with Processing. By the end of the class, he reported that, although it seemed difficult, he liked programming and would want to continue to learn.

5.3.5 Recommendations

Recommendations for future courses at Juvenile Hall would be to plan early, plan for flexibility, and have lots of volunteers. In addition, it is beneficial to incorporate multiple subjects to adapt to different students' course requirement needs. We also found an effective class setup to be a short lecture followed by a short assignment. If there was still time for another iteration of lecture/assignments we would continue once students were finished.

5.3.6 Threats to Validity

One of the threats to validity was unexpectedly meeting the students prior to the first day of class. Though it was helpful to meet them and introduce the course topic prior to the first lesson, this was an unplanned event and inflated student

responses on the pre-survey; by the time we gave them the pre-survey asking their opinions on computer science, they had already been informed of some benefits of computer science. However, although skewing the data, meeting the students early was beneficial and made the first class easier.

Another threat to the validity is the sample size. There were only a total of seven students. This means our survey results are not strong forms of evidence nor are they statistically valid. Furthermore, only *three* of the students took both the pre-and-post surveys for comparison. However, the focus of this study was not quantitative results. The value was in pioneering a new course and making the connections to make it sustainable. Future work could include a heavier focus on quantitative analysis once this course is more established.

Finally, although we instructed the students to keep the pre and post surveys anonymous, some students wrote their name on the post-survey. Since they chose to disclose their name, they may not have answered as honestly.

Chapter 6

CONCLUSION AND FUTURE WORK

In conclusion, our research explores how to improve and spread K-12 CS curriculum through creativity and choice. We explored creative choice in a fifth grade curriculum and created an introductory CS course for Coastal Valley Academy students at San Luis Obispo County’s Juvenile Hall.

6.1 Fifth Grade Curriculum

The fifth grade curriculum consisted of two interventions. The first gave each student a creative and non-creative assignment and gathered their opinions and preferences. The second reported on effects of creativity on students learning new material, analyzing the impact of a creative versus non-creative treatment on students’ efficacy, performance, and engagement.

Overall, our research found that students do value creative choice and freedom in their assignments, as well as a sense of ownership, picking, and choice. In addition, creativity played a large role in swaying overall preferences on worksheets. Conversely, we found that the students that felt they were struggling in the lab preferred the clarity of non-creative worksheets.

We also explored how students engaged with the varying degrees of creative freedom offered to them. We found that most students reported to like the creative freedom of multiple choice assignments, but that very few students actually chose to make their own options if given suggestions to choose from.

Finally, we found that our creative assignments had no impact on efficacy or performance.

Considering these findings, we recommend a limited form of creativity on future worksheets.

6.1.1 Future Work

We would dedicate future work to exploring the degrees of creative freedom within the multiple choice style worksheet, the multiple choice style creativity versus other styles of creative worksheets, and the effects of these variants on student opinion, efficacy, and performance.

In future work, we would like to continue refining the multiple choice style of creative worksheets. We hypothesize that restricting some of the options on them, providing limited creativity, would still give students a sense of ownership while providing clarity and guidance for struggling students. We found multiple choice is an effective method of providing creative assignments because it does provide guided choice, giving both structure and freedom. Students reported to like it and to feel they had ownership despite its structure. In addition, some students report being confused if given too much freedom, indicating the benefits of limited creativity. We would like to see if students that struggled with the worksheets would find them more understandable without option ‘D’ (create your own) and if the students that liked the creative assignments would still have the same sense of freedom without this option. We would like to test this and other degrees of creative freedom.

In addition to degrees of creativity, we would like to further look into student perceptions across different styles of creativity. While our work attempted to report on student opinions of creative assignments, it only provided feedback on three sets of

creative worksheets. These worksheets were of only a few creative styles (guided creativity in *Intervention One* and both free form and guided creativity in *Intervention Two*). Future work would be to rigorously validate and compare different styles of creative assignments and see which ones students prefer and which aspects they like of each.

Overall, future work would be to validate the effects of different degrees and styles of freedom and creativity. We would like to see the effects of removing the free-form option in *Intervention One*. We also would like to compare different types of creativity - our multiple choice options for many elements versus Bootstraps' unlimited options for four aspects (mentioned in *Related Works*). We also would like to look into maximizing a sense of ownership with limited confusion and actual freedom to aid in moving curriculum forward while keeping the benefits of creativity. *Intervention Two* had a different way of presenting creative options (more free form instead of the multiple choice of this sections), and thus we began to address different styles of creativity there. Finally, our initial responses found that creative assignments had no negative impacts on efficacy and performance, but in the future we would like to continue looking at these effects against different degrees of freedom and differing styles of creativity.

6.2 Juvenile Hall Curriculum

In the second half of this thesis, we created an introductory CS course for CVA students at San Luis Obispo County's Juvenile Hall. This course attempted to empower students that previously did not have access to CS courses through creative, project-based curriculum. We designed our class around 2D game-design, with an emphasis on creativity, course flexibility, and tutoring-style work sessions. We found the course

to have initial success, as all students reported to like the class and a desire to continue to learn computer programming. The students also reported liking the game-design aspect of the course and that they wish the course was longer.

Although the course seemed successful, more vigorous study would be needed to experimentally validate our initial findings. For example, while we intentionally incorporated creativity into the curriculum, we were not able to specifically investigate the effects of varying degrees of creativity, as we attempted to do with the fifth graders. Now that the class has been established and has shown initial success, we would like to begin experimenting to find the most effective way to teach this class.

BIBLIOGRAPHY

- [1] About processing.js. <http://processingjs.org/>. Accessed: 2019-05-13.
- [2] About restorative partners.
<https://www.restorativepartners.org/about.php>. Accessed:
2019-02-08.
- [3] Bootstrap: equity, scale, rigor. <https://www.bootstrapworld.org/>. Accessed:
2019-05-11.
- [4] Cal Poly Github. <http://www.github.com/CalPoly>.
- [5] Code.org: About us. <https://code.org/about>. Accessed: 2019-05-12.
- [6] Code.org and diversity in computer science. <https://code.org/diversity>.
Accessed: 2019-01-29.
- [7] Data usa: Software engineering.
<https://datausa.io/profile/cip/140903/#demographics>. Accessed:
2018-01-29.
- [8] Probation: County of san luis obispo.
<https://www.slocounty.ca.gov/Departments/Probation/Juvenile-Hall.aspx>. Accessed: 2019-05-27.
- [9] processing.js. <https://valentin.dasdeck.com/processing/>. Accessed:
2019-02-08.
- [10] Us and world population clock. <https://www.census.gov/popclock/>.
Accessed: 2018-01-29.

- [11] E. Aivaloglou and F. Hermans. Early programming education and career orientation: The effects of gender, self-efficacy, motivation and stereotypes. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, pages 679–685, New York, NY, USA, 2019. ACM.
- [12] M. Armoni, O. Meerbaum-Salant, and M. Ben-Ari. From scratch to “real” programming. *Trans. Comput. Educ.*, 14(4):25:1–25:15, Feb. 2015.
- [13] A. Bandura. Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2):191 – 215, 1977.
- [14] A. Bandura. Guide for constructing self-efficacy scales. *Self-efficacy beliefs of adolescents*, 5(1):307–337, 2006.
- [15] A. Bandura, N. E. Adams, and J. Beyer. Cognitive processes mediating behavioral change. *Journal of Personality and Social Psychology*, 35(3):125 – 139, 1977.
- [16] A. Bandura and D. H. Schunk. Cultivating competence, self-efficacy, and intrinsic interest through proximal self-motivation. *Journal of Personality and Social Psychology*, 41(3):586 – 598, 1981.
- [17] A. Bandura and S. Wessels. Self-efficacy, 1994.
- [18] G. V. Caprara, R. Fida, M. Vecchione, G. Del Bove, G. M. Vecchio, C. Barbaranelli, and A. Bandura. Longitudinal analysis of the role of perceived self-efficacy for self-regulated learning in academic continuance and achievement. *Journal of Educational Psychology*, 100(3):525 – 534, 2008.
- [19] L. B. Cassel, A. McGettrick, M. Guzdial, and E. Roberts. The current crisis in

- computing: What are the real issues? *SIGCSE Bull.*, 39(1):329–330, Mar. 2007.
- [20] L. J. Charleston, P. L. George, J. F. L. Jackson, J. Berhanu, and M. H. Amechi. Navigating underrepresented stem spaces: Experiences of black women in us computing science higher education programs who actualize success. *Journal of Diversity in Higher Education*, 7(3):166 – 176, 2014.
- [21] L.-Y. Chen, B. Hsiao, C.-C. Chern, and H.-G. Chen. Academic self-efficacy scale. *PsycTESTS*, 2014.
- [22] S. Cheryan, V. C. Plaut, P. G. Davies, and C. M. Steele. Ambient belonging: How stereotypical cues impact gender participation in computer science. *Journal of Personality and Social Psychology*, 97(6):1045 – 1060, 2009.
- [23] S. Cheryan, S. A. Ziegler, A. K. Montoya, and L. Jiang. Why are some stem fields more gender balanced than others?. *Psychological Bulletin*, 143(1):1 – 35, 2017.
- [24] J. P. Cohoon. An introductory course format for promoting diversity and retention. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE ’07, pages 395–399, New York, NY, USA, 2007. ACM.
- [25] K.-. C. S. F. S. Committee. K-12 computer science framework. Technical report, New York, NY, USA, 2016.
- [26] E. L. Cowen, W. C. Work, A. D. Hightower, P. A. Wyman, G. R. Parker, and B. S. Lotyczewski. Perceived self-efficacy scale. *PsycTESTS*, 1991.
- [27] D. H. Cropley. Promoting creativity and innovation in engineering education. *Psychology of Aesthetics, Creativity, and the Arts*, 9(2):161 – 171, 2015.

- [28] K. M. Davis, Z. Wood, and J. Wilcox. Eighteen hours of code with fifth grade students (abstract only). In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, SIGCSE '16, pages 694–694, New York, NY, USA, 2016. ACM.
- [29] J. Q. Dawson, M. Allen, A. Campbell, and A. Valair. Designing an introductory programming course to improve non-majors' experiences. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, pages 26–31, New York, NY, USA, 2018. ACM.
- [30] C. I. Fertman and B. A. Primack. Elementary student self efficacy scale. *PsycTESTS*, 2009.
- [31] C. I. Fertman and B. A. Primack. Elementary student self efficacy scale development and validation focused on student learning, peer relations, and resisting drug use. *PubMed*, 39:23–38, 2009.
- [32] D. Franklin, G. Skifstad, R. Rolock, I. Mehrotra, V. Ding, A. Hansen, D. Weintrop, and D. Harlow. Using upper-elementary student performance to understand conceptual sequencing in a blocks-based curriculum. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '17, pages 231–236, New York, NY, USA, 2017. ACM.
- [33] M. Gambin and M. wicicka. Self-efficacy scale for children. *PsycTESTS*, 2012.
- [34] M. S. Gunbatar. Examination of undergraduate and associate degree students' computer programming attitude and self-efficacy according to thinking style, gender and experience. *Contemporary Educational Technology*, 9(4):354 – 373, 2018.

- [35] M. Habibi, K. Tahmasian, and L. Ferrer-Wreder. Self-efficacy questionnaire for children–persian version. *PsycTESTS*, 2014.
- [36] A. K. Hansen, H. A. Dwyer, A. Iveland, M. Talesfore, L. Wright, D. B. Harlow, and D. Franklin. Assessing children’s understanding of the work of computer scientists: The draw-a-computer-scientist test. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE ’17, pages 279–284, New York, NY, USA, 2017. ACM.
- [37] A. K. Hansen, A. Iveland, C. Carlin, D. B. Harlow, and D. Franklin. User-centered design in block-based programming: Developmental & pedagogical considerations for children. In *Proceedings of the The 15th International Conference on Interaction Design and Children*, IDC ’16, pages 147–156, New York, NY, USA, 2016. ACM.
- [38] B. A. Hennessey. If i were secretary of education: A focus on intrinsic motivation and creativity in the classroom. *Psychology of Aesthetics, Creativity, and the Arts*, 9(2):187 – 192, 2015.
- [39] D. Henriksen, M. Henderson, E. Creely, S. Ceretkova, M. Černochová, E. Sendova, E. T. Sointu, and C. H. Tienken. Creativity and technology in education: An international perspective. *Technology, Knowledge and Learning*, 23(3):409–424, Oct 2018.
- [40] C. J. Infurna, D. Riter, and S. Schultz. Factors that determine preschool teacher self-efficacy in an urban school district. *International Electronic Journal of Elementary Education*, 11(1):1 – 7, 2018.
- [41] J. Jinks and V. Morgan. Children’s perceived academic self-efficacy: An inventory scale. *The Clearing House*, 72(4):224–230, 1999.

- [42] J. P. P. Jokinen. Technology coping dimensions questionnaire. *PsycTESTS*, 2015.
- [43] E. Kazakoff and M. Bers. Programming in a robotics context in the kindergarten classroom : The impact on sequencing skills. In *Journal of Educational Multimedia and Hypermedia*, volume 21, pages 371–391, 2012.
- [44] P. Kinnunen and L. Malmi. Cs minors in a cs1 course. In *Proceedings of the Fourth International Workshop on Computing Education Research, ICER ’08*, pages 79–90, New York, NY, USA, 2008. ACM.
- [45] B. Lohse, L. Cunningham-Sabo, L. M. Walters, and J. E. Stacey. Children’s cooking self-efficacy measure. *PsycTESTS*, 2011.
- [46] A. M. Mahasneh and A. F. Alwan. The effect of project-based learning on student teacher self-efficacy and achievement. *International Journal of Instruction*, 11(3):511 – 524, 2018.
- [47] H. W. Marsh, R. Pekrun, P. D. Parker, K. Murayama, J. Guo, T. Dicke, and A. K. Arens. The murky distinction between self-concept and self-efficacy: Beware of lurking jingle-jangle fallacies. *Journal of Educational Psychology*, 111(2):331 – 353, 2019.
- [48] A. Master, S. Cheryan, and A. N. Meltzoff. Computing whether she belongs: Stereotypes undermine girls interest and sense of belonging in computer science. *Journal of Educational Psychology*, 108(3):424 – 437, 2016.
- [49] A. Mejia, A. Filus, R. Calam, A. Morawska, and M. R. Sanders. Child adjustment and parent efficacy scale–spanish version. *PsycTESTS*, 2016.
- [50] T. Migler. Jail classes. <https://theresamigler.com/jail-classes/>. Accessed: 2019-02-08.

- [51] L. T. O'Brien, D. M. Garcia, A. Blodorn, G. Adams, E. Hammer, and C. Gravelin. An educational intervention to improve womens academic stem outcomes: Divergent effects on well-represented vs underrepresented minority women. *Cultural Diversity and Ethnic Minority Psychology*, 2019.
- [52] O. Ozyurt. An analysis on distance education computer programming students' attitudes regarding programming and their self-efficacy for programming. *Turkish Online Journal of Distance Education*, 16(2):111 – 121, 2015.
- [53] C. Pastorelli, G. V. Caprara, C. Barbaranelli, J. Rola, S. Rozsa, and A. Bandura. The structure of children's perceived self-efficacy: A cross-national study. *European Journal of Psychological Assessment*, 17(2):87 – 97, 2001.
- [54] M. Peker and R. Erol. Investigation of the teacher self-efficacy beliefs of math teachers. *Malaysian Online Journal of Educational Sciences*, (6):1 – 11, 2018.
- [55] P. R. Pintrich, O. And, N. C. for Research to Improve Postsecondary Teaching, and M. Learning, Ann Arbor. A manual for the use of the motivated strategies for learning questionnaire (mslq). 1991.
- [56] A. Quade. Development and validation of a computer science self-efficacy scale for cs0 courses and the group analysis of cs0 student self-efficacy. In *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing*, pages 60–64, April 2003.
- [57] K. Rainey, M. Dancy, R. Mickelson, E. Stearns, and S. Moller. Race and gender differences in how sense of belonging influences decisions to major in stem. *International Journal of STEM Education*, 5(1):10, Apr 2018.

- [58] D. Reed. Rethinking cs0 with javascript. *SIGCSE Bull.*, 33(1):100–104, Feb. 2001.
- [59] D. Riley. Rigor/us: Building boundaries and disciplining diversity with standards of merit. *Engineering Studies*, 9(3):249–265, 2017.
- [60] L. Ritchie and A. Williamon. Self-efficacy for musical learning—children’s version. *PsycTESTS*, 2011.
- [61] W. Robinson. From scratch to patch: Easing the blocks-text transition. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education*, WiPSCE ’16, pages 96–99, New York, NY, USA, 2016. ACM.
- [62] E. Schanzer, K. Fisler, and S. Krishnamurthi. Assessing bootstrap: Algebra students on scaffolded and unscaffolded word problems. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE ’18, pages 8–13, New York, NY, USA, 2018. ACM.
- [63] E. Schanzer, S. Krishnamurthi, and K. Fisler. Creativity, customization, and ownership: Game design in bootstrap: Algebra. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE ’18, pages 161–166, New York, NY, USA, 2018. ACM.
- [64] E. Schofield, M. Erlinger, and Z. Dodds. Mycs: Cs for middle-years students and their teachers. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE ’14, pages 337–342, New York, NY, USA, 2014. ACM.
- [65] O. Sensoy and H. I. Yildirim. Impact of technological pedagogical content knowledge based education applications on prospective teachers’ self-efficacy belief levels toward science education. *Journal of Education and Training Studies*, 6(10):29 – 38, 2018.

- [66] R. J. Sternberg. A triangular theory of creativity. *Psychology of Aesthetics, Creativity, and the Arts*, 12(1):50 – 67, 2018.
- [67] L. K. Teck. Game design workshop: Cultivating creativity from constraint. In *SIGGRAPH Asia 2017 Symposium on Education*, SA '17, pages 10:1–10:3, New York, NY, USA, 2017. ACM.
- [68] M.-J. Tsai, C.-Y. Wang, and P.-F. Hsu. Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research*, 56(8):1345 – 1360, 2019.
- [69] J. Tsan, K. E. Boyer, and C. F. Lynch. How early does the cs gender gap emerge?: A study of collaborative problem solving in 5th grade computer science. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, SIGCSE '16, pages 388–393, New York, NY, USA, 2016. ACM.
- [70] C. Wilcox and A. Lionelle. Quantifying the benefits of prior programming experience in an introductory computer science course. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, pages 80–85, New York, NY, USA, 2018. ACM.
- [71] Z. Wood. Fifth grade introduction to computer programming using processing - pcs. <http://users.csc.calpoly.edu/~zwood/Outreach/PCS.html>. Accessed: 2019-02-08.
- [72] D. Xu, A. Cadle, D. Thompson, U. Wolz, I. Greenberg, and D. Kumar. Creative computation in high school. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, SIGCSE '16, pages 273–278, New York, NY, USA, 2016. ACM.

- [73] H.-L. Yang and H.-H. Cheng. Skills and self-efficacy of information systems developers questionnaire. *PsycTESTS*, 2009.
- [74] S. Yardi and A. Bruckman. What is computing?: Bridging the gap between teenagers’ perceptions and graduate students’ experiences. In *Proceedings of the Third International Workshop on Computing Education Research*, ICER ’07, pages 39–50, New York, NY, USA, 2007. ACM.
- [75] L. Ye, A. Posada, and Y. Liu. The moderating effects of gender on the relationship between academic stress and academic self-efficacy. *International Journal of Stress Management*, 25(S1):56 – 61, 2018.
- [76] Z. Zhou. What cognitive neuroscience tells us about creativity education: A literature review. *Global Education Review*, 5(1):20 – 34, 2018.
- [77] K. J. Zullig, D. A. Teoli, and R. F. Valois. Self-efficacy questionnaire for children–revised. *PsycTESTS*, 2011.

APPENDICES

Appendix A

IRB PROPOSAL

Below is the material concerning the 5th grade interventions sent to the university's Institutional Review Board for approval. Included are the: proposal (with links to curriculum and surveys), permission forms, and script on what to tell the children before distributing the surveys.

Title

Evaluating Creative Choice in K-12 Computer Science Curriculum

Primary Investigator and Advisor

PI: Kirsten Mork, Computer Science Department, Cal Poly San Luis Obispo

Advisor: Zoë Wood, Computer Science Department, Cal Poly San Luis Obispo

Statement of Purpose/Summary

The purpose of this study is to discover more effective ways to teach Computer Science in K-12 settings, specifically by evaluating how creative choice can improve learning and self-efficacy. The study will be conducted in a 5th grade computer science class which has existing curriculum taught by a fifth grade teacher. I will also teach 2-6 weeks of modified curriculum designed to evaluate the effects of creative choice. There are no known risks, but benefits include improving computer science curriculum and increasing student self-efficacy. The data will be collected through anonymous surveys and through anonymous results on problems within the curriculum.

MethodsSubjects and Subject Characteristics

The subjects are fifth grade students at Peabody Charter School in Santa Barbara. There are four classes of about 25 students.

Investigator

Kirsten Mork. Has experience teaching and working with kids through volunteering at after school computer science programs, years working as a summer camp counselor, and teaching Sunday School for elementary students.

Materials and Procedures

The subjects are already attending a computer lab for an hour each week and are already being taught computer science curriculum as a part of their regular schooling. They are used to visitors helping to teach the material as well.

I am working with the fifth grade computer science teacher and the school's principal to see if I can give the practice problems and teach the modified curriculum. I will also get the teacher's approval on the parental permission form (provided in English and Spanish) and send the form home with the student's to get permission to administer the surveys.

The study is split into two sections: 1) providing practice problems to assess creative choice, and 2) teaching curriculum with varying degrees of creative choice. A short, open-ended survey will be given after part 1, and pre and post surveys will be given for both the existing and additional curriculum for part 2. The links for the survey drafts are below. In addition, the modified curriculum is attached later in the IRB manager project submission or included below:

- Part 1 Survey:
https://docs.google.com/forms/d/e/1FAIpQLSdNpx_dSsn6_HkbSjaTnW-RNKqCdRDeS8xZJIsPHiOA_lyPWg/viewform?usp=sf_link
- Pre/Post Survey on Efficacy:
https://docs.google.com/forms/d/e/1FAIpQLSdsMnurAHt2yhZjvgQCZIJkfZYzEQIXRHCE0xDiEnJ8rOmC7Q/viewform?usp=sf_link
- Pre/Post Survey on Computer Anxiety:
https://docs.google.com/forms/d/e/1FAIpQLSefj2ely1nnqPwMz3EgLBngu5iRr-pLZsY2qzIqR4tufdbSag/viewform?usp=sf_link
- Curriculum:
<https://docs.google.com/document/d/10C-aKwt9JQaHfXT3u-yzO--vietOYWptIHJE33eSK8c/edit?usp=sharing>

There are no risks, as the students already participate in the computer science course as a part of their schooling, so there is already procedure for students getting to and from the classroom. The computer class is an hour long, and will be split between about 20 minutes of instruction and 40 minutes of students doing practice problems. The surveys will be given at the beginning and/or end of the practice problem section.

The surveys will be identified by the last four digits of an ID and will at no point be associated with their name. My advisor and I are the only ones who will have access to the surveys, which are stored in my private google-drive account.

The parents will be informed of the details and purpose of the surveys in the permission form, and the students will additionally be told what the purpose is at the beginning of the lessons and surveys.

Study Location

The setting, as mentioned above, is a computer science lab at Peabody Elementary school. The students already attend the lab for an hour once a week and have procedure for getting to and from the lab. As mentioned above, the surveys will be administered as an online google-survey here. The links are listed in the “Materials and Procedures” section. The student’s will be told the purpose of the survey and the parents and students both know it is optional and can choose to not give consent.

Informed Consent Forms

The parental consent form and scripts for student consent are included below:

Parent Permission Form (English)

Parental/Guardian Permission Form

INFORMED PERMISSION TO PARTICIPATE IN A RESEARCH PROJECT
Evaluating Creative Choice in K-12 Computer Science Curriculum

A research project on teaching computer science to K-12 students is being conducted by Kirsten Mork, a graduate student in the Department of Computer Science at Cal Poly, San Luis Obispo, under the supervision of Dr. Zoë Wood. The purpose of the study is to teach and evaluate programming curriculum and see if giving students creative choice improves their learning and confidence.

Your child is being asked to take part in this study by learning about programming and answering a few survey questions before and after. Their participation will take place during the computer science lab they attend on Monday/Thursdays. Please be aware that they are not required to answer the survey questions in order to participate in the program and you or they may discontinue their participation at any time. Your child's confidentiality will be protected and the research results will not include their name or identifiable information. There are no known risks anticipated with this study. Potential benefits associated with the study are teaching your child programming and improving future curriculum. If you or your child have questions regarding this study or would like to be informed of the results when the study is completed, please feel free to contact Kirsten Mork at 925-895-9978 or klmork@calpoly.edu. If you or your child have concerns regarding the manner in which the study is conducted, you may contact Dr. Michael Black, Chair of the Cal Poly Institutional Review Board, at (805) 756-2894, mblack@calpoly.edu, or Ms. Debbie Hart, Compliance Officer, at (805) 756-1508, dahart@calpoly.edu. If you agree to allow your child to voluntarily participate in this research project as described, please indicate your agreement by signing below. Thank you for your participation in this research.

Name(s) of Child/Children/Dependent(s) Involved in this Research:

Signature of Parent/Legal Guardian

Date

Signature of Researcher

Date

Parent Permission Form (Spanish)

(Note For IRB: This form was translated by a conversational Spanish speaker and reviewed by a native Spanish speaker)

Formulario de Inclusión Para Padres / Tutores

INFORMACIÓN DE ELECCIÓN PARA PARTICIPAR EN UN PROYECTO DE INVESTIGACIÓN
Evaluación De La Elección Creativa En El Plan De Estudios En Ciencias De Computación
Para Grados K-12

Kirsten Mork, un estudiante graduado en el Departamento de Ciencias de Computación en Cal Poly, San Luis Obispo, está llevando a cabo un proyecto de investigación sobre la enseñanza de ciencias de computación a estudiantes de grados K-12, bajo la supervisión de la Dra. Zoë Wood. El propósito del estudio es evaluar la elección creativa en el currículo de programación de computadoras.

Como parte del plan de estudios del laboratorio de computación de quinto grado existente, se le pedirá a su hijo que participe en una breve encuesta sobre su actitud y experiencia antes y después de dos o cuatro de sus lecciones de laboratorio de computación. Su participación se llevará a cabo durante el laboratorio de ciencias de computación al que asisten los lunes o los jueves. No se requiere que los niños respondan las preguntas de la encuesta para participar en las lecciones de esas semanas, y usted o ellos pueden interrumpir su participación en la encuesta en cualquier momento.

Se protegerá la confidencialidad de su hijo (no se les preguntará a los niños por su nombre) y los resultados de la investigación no incluirán ningún nombre, o información identificable. No hay riesgos conocidos o anticipados con la participación en la encuesta. Los beneficios potenciales asociados con el estudio es para mejorar el futuro plan de estudios del laboratorio de computación. Si usted o su hijo tienen preguntas sobre este estudio, o desean recibir información sobre los resultados cuando se complete el estudio, no dude en comunicarse con Kirsten Mork al 925-895-9978, klmork@calpoly.edu. Si usted o su hijo tienen inquietudes acerca de la manera en que se realiza el estudio, puede comunicarse con el Dr. Michael Black, Presidente de la Junta de Revisión Institucional de Cal Poly, al (805) 756-2894, mblack@calpoly.edu, o Sra. Debbie Hart, Oficial de Cumplimiento, al (805) 756-1508, dahart@calpoly.edu.

Si desea que su hijo participe voluntariamente en este proyecto de investigación como se describe, indique su acuerdo firmando abajo. Gracias por su participación en esta investigación.

Nombre(s) del Niño / Hijos / Dependiente(s) que se incluirán en esta investigación:

Firma del Padre / Madre / Tutor Legal

Fecha

Child Consent Script

Hello everyone. Today I am going to give some worksheets and practice problems for you to work on, just like you normally do in this lab. This time though, I also am working on a project for my college. I'm trying to see what is the most helpful way to teach this class - I want to see if giving more creativity in the assignments is good or not. Do you mind helping me with this project? If you'd like to help, you can fill out a survey before to tell me how you feel about programming. After the lessons I have another survey to ask you what you liked and didn't like about the lessons and to see if any of your opinions have changed.

The surveys are anonymous and it won't affect your grade. If you decide you don't want to you can stop at any time.

Appendix B

IRB MODIFICATION REQUEST

The initial IRB protocol was sent before the details for the Juvenile Hall class were worked out. Below is the material for adding the Juvenile Hall class to the IRB request. The Modification Request and additional Juvenile Hall Permission Form are included below.

Type of Modifications Requested

Change in Location

Change in Study Population

Description of Proposed Changes

The proposed changes include broadening the research to include a new demographic. We are not proposing to change any of the work involving the 5th grade students, and still plan to continue working with them as previously was approved by the IRB. The changes are to extend the curriculum being taught to the fifth graders and teach a similar class at Juvenile Hall and evaluate its success. The process will be very similar to how we are working with the fifth grade students.

I am already cleared to enter Juvenile Hall through the organization Restorative Partners. I have pitched teaching computer science class to Juvenile Hall supervisors; they are very supportive of launching this new class and want the class incorporated into the youths' school program. I will get approval from Juvenile Hall on the curriculum and surveys before proceeding.

The idea is to teach the class at the beginning of 2019. There will be an anonymous pre and post survey to evaluate the success of the class. We will also evaluate student comprehension by evaluating work throughout the course. There is no risk since the surveys are completely anonymous.

There will be at most 7 students in CVA (the part of Juvenile Hall I would be working in) at the time of the class. The materials and computers are provided by Juvenile Hall and the class would be taught in one of the classrooms the students normally use for school. Everything will be cleared by Juvenile Hall - materials and surveys and curriculum - before proceeding.

We also have a permission form giving consent to use the results in our research. This will be given to Juvenile Hall to sign before the class and signed for each of the students at that time. Overall, there is no risk since Juvenile Hall must approve the entire process to allow me to teach, and also all reported results are anonymous. Benefits include empowering Juvenile Hall students to pursue computer science by teaching problem solving skills and basic computing concepts.

We believe it is important to explore creating and assessing introductory computer science curriculum for incarcerated youth because while 58% of all new jobs in STEM are in computing, only 35% of US high schools teach any computer science courses, and access to computer science education is even less for the incarcerated youth population. In addition, in general students who learn computer science in high school are 6 times more likely to major in it, thus exposing students at this time in their life to computing has the potential to impact their future decisions and job opportunities.

However, despite the importance of CS education for high school students, there has been quite a bit of evidence of racial, socioeconomic, and gender inequalities in CS educations. To close this gap, it is important to consider the following: creating equal access in distribution, building equity into the curriculum itself, and designing age appropriate material. In order to achieve equality in Computer Science curriculum, we must make sure the curriculum is not only being taught to all populations, but tailored to their specific needs so it is taught in an accessible way. This study strives to bridge the gap by addressing both points for one of the less reached groups. Given that the incarcerated youth population has very limited access to computer science education and that this education has benefits for this population, we believe it is important to introduce this curriculum to this population and study the effectiveness of this computing curriculum. In particular, we want to make sure the CS curriculum being used is engaging for the students participating in the study (i.e. the incarcerated youth population at juvenile hall).

Guardian Permission Form

Attached below:

Parental/Guardian Permission Form

INFORMED CHOICE TO PARTICIPATE IN A RESEARCH PROJECT Evaluating Creative Choice in K-12 Computer Science Curriculum

A research project on teaching computer science to K-12 students is being conducted by Kirsten Mork, a graduate student in the Department of Computer Science at Cal Poly, San Luis Obispo, under the supervision of Dr. Zoë Wood. The purpose of the study is to evaluate creative choice in computer programming curriculum.

As a part of the existing fifth grade computer lab curriculum, your child will be asked to take part in a short survey about their attitude and experience before and after two-four of their computer lab lessons. Their participation will take place during the computer science lab they attend on either Mondays or Thursdays. Children are not required to answer the survey questions in order to participate in the lessons those weeks, and you or they may discontinue their participation in the survey at any time.

Your child's confidentiality will be protected (children are not asked for their name) and the research results will not include any names or identifiable information. There are no known risks anticipated with participating in the survey. Potential benefits associated with the study are improving future computer lab curriculum. If you or your child have questions regarding this study or would like to be informed of the results when the study is completed, please feel free to contact Kirsten Mork at 925-895-9978 or klmork@calpoly.edu. If you or your child have concerns regarding the manner in which the study is conducted, you may contact Dr. Michael Black, Chair of the Cal Poly Institutional Review Board, at (805) 756-2894, mblack@calpoly.edu, or Ms. Debbie Hart, Compliance Officer, at (805) 756-1508, dahart@calpoly.edu.

If you give your child permission to voluntarily participate in this research project as described, please indicate your agreement by signing below. Thank you for your participation in this research.

Name(s) of Child/Children/Dependent(s) to be included in this research:

Signature of Parent/Legal Guardian

Date

Appendix C

PEABODY-INTERVENTION 1: CC MOUSE

The following is the curriculum for intervention 1 (testing student preferences concerning creative choice while practicing concepts). This is the set of worksheets where students could use creative choice to create a mouse.

ID: _____

Practice Problems-Processing Basics

J. Wilcox, Z. Wood, K. Mork

Today we are going to do some practice by creating a mouse! This week, choose which options you like best or create your own options as you follow the steps.

➤ First, let's copy the grid code to help draw our mouse.

Google search: **wilcox peabody processing**

Click on: **Base Code for grid**




Click on the page and type the command to select all: **cnt-a**

Type the command to copy: **cnt-c**

Go to the processing editor and click on the screen and type the command to paste: **cnt-v**

➤ **For each of the following, circle the option you pick** - if you create your own be sure to **write the numbers in for the command you chose.**

➤ Next, let's pick a background color to place our mouse in. Pick from the following or make up your own. Find the line "background(255);" in the code you pasted and **replace it with your choice.**

A. Light Blue  background(0, 200, 250);	B. Dark Blue  background(17, 30, 108);
C. Light Purple  background(150, 111, 214);	D. background(__, __, __);

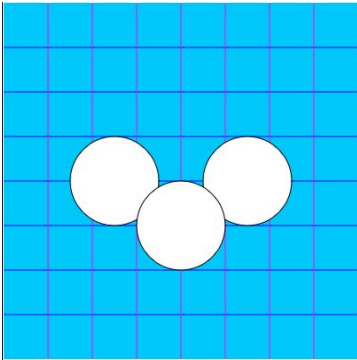
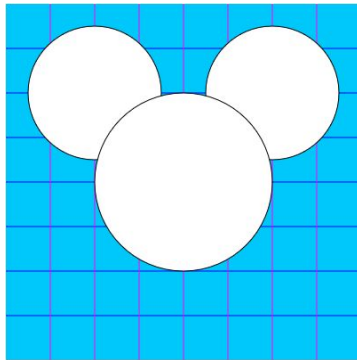
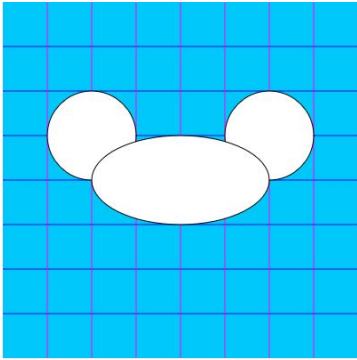
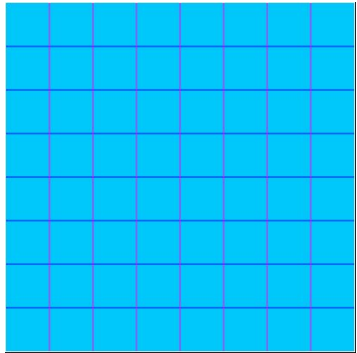
➤ Now let's pick an outline color for our mouse. Type one of the following:

A. stroke(0); //black	B. stroke(0, 255, 0); //green
C. noStroke(); //no outline	D. stroke(__, __, __); //other




➤ Next, we will create the mouse head. Circle the color you want or write in your own:

A. <input type="checkbox"/> fill(255); //white	B. <input type="checkbox"/> fill(50); //gray
C. <input type="checkbox"/> fill(121, 56, 2); //brown	D. fill(____,____,____); //other

➤ Select one of the following or create your own head for option D. You can use the grid on the next page to help you design a head if you would like. If you choose option D, you will have to keep designing the rest of your mouse too since the example features might not fit the size you choose. Circle the option you chose and type:

<p>A. Head 1</p>  <pre>ellipse(125, 200, 100, 100); ellipse(275, 200, 100, 100); ellipse(200, 250, 100, 100);</pre>	<p>B. Head 2</p>  <pre>ellipse(100, 100, 150, 150); ellipse(300, 100, 150, 150); ellipse(200, 200, 200, 200);</pre>
<p>C. Head 3</p>  <pre>ellipse(100, 150, 100, 100); ellipse(300, 150, 100, 100); ellipse(200, 200, 200, 100);</pre>	<p>D. Other</p>  <pre>ellipse(____, ____, ____, ____); ellipse(____, ____, ____, ____); ellipse(____, ____, ____, ____);</pre>

➤ Select a color for the nose!

A.  fill(254, 127, 156); //pink	B.  fill(249, 166, 2); //orange
C.  fill(57, 255, 20); //light green	D. fill(____,____,____); //other

➤ Now, go to the option for the head you picked.

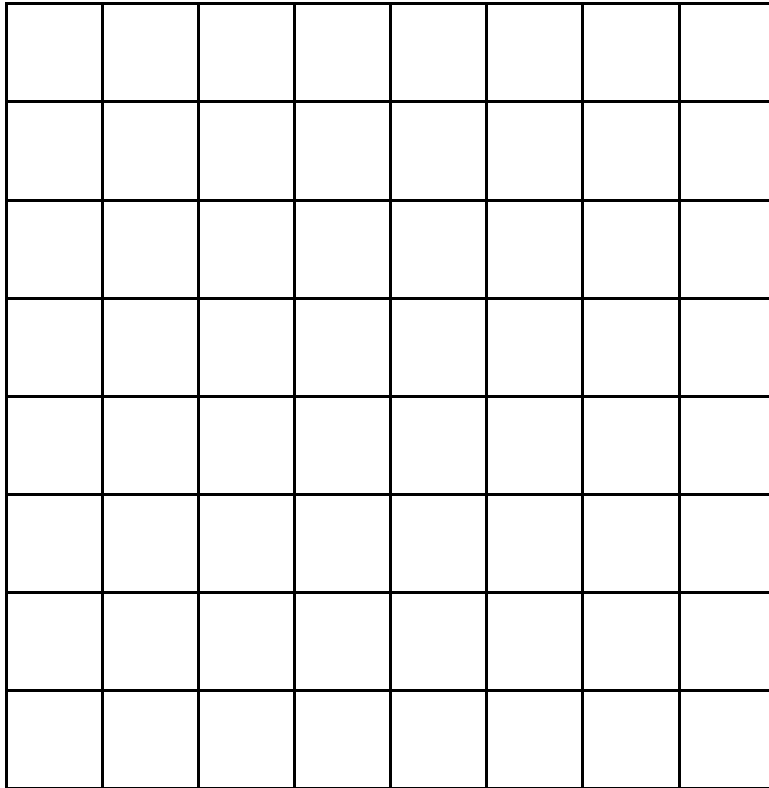
A: go to page 4

B: go to page 7

C: go to page 10

D: go to page 13

(0,0)



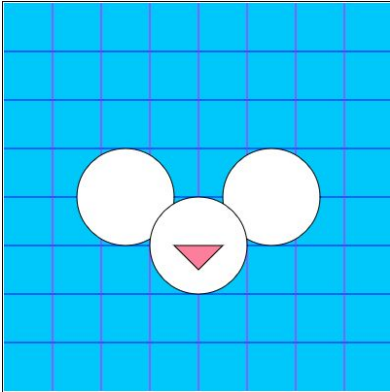
(400, 400)

A. Head 1 Options

For each of the following, circle the option you picked. If you create your own, fill in the command with the numbers you used!

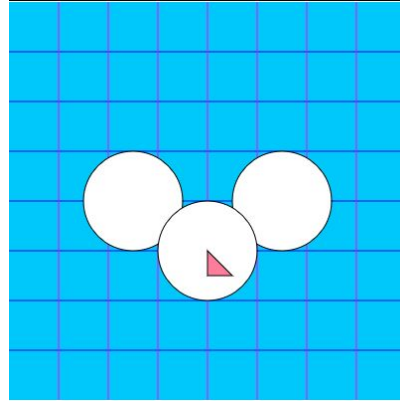
➤ Pick a nose from the following options, or create your own.

A. Nose 1



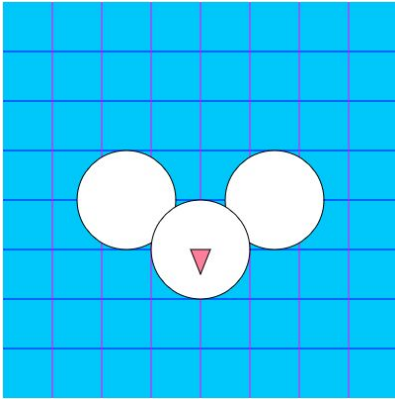
triangle(200, 275, 175, 250, 225, 250);

B. Nose 2



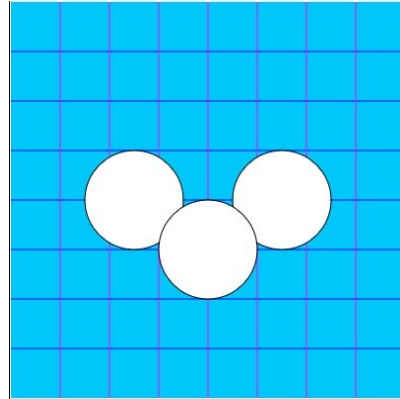
triangle(200, 275, 200, 250, 225, 275);

C. Nose 3



triangle(200, 275, 190, 250, 210, 250);

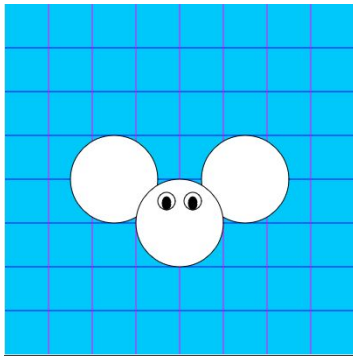
D. Other



triangle(____, ____, ____, ____, ____, ____);

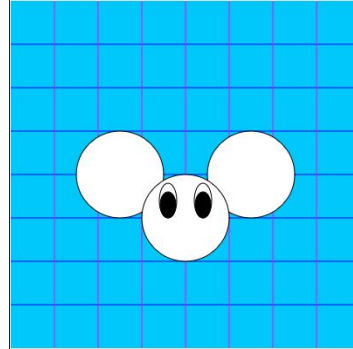
➤ Now we will draw the eyes. Circle one of the following options, or create your own.

A. Eyes 1



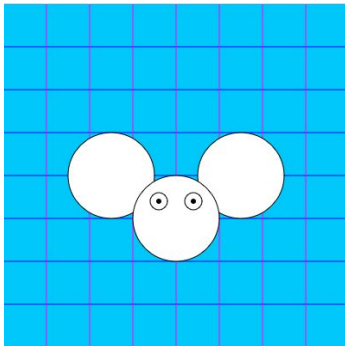
```
//Draw the outer circle
fill(255);
ellipse(185, 225, 20, 20);
ellipse(215, 225, 20, 20);
//draw the pupil
fill(0);
ellipse(185, 227.5, 10, 15);
ellipse(215, 227.5, 10, 15);
```

B. Eyes 2



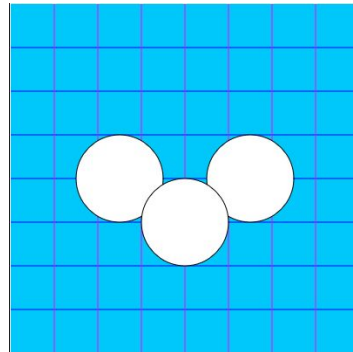
```
//Draw the outer circle
fill(255);
ellipse(180, 230, 20, 40);
ellipse(220, 230, 20, 40);
//draw the pupil
fill(0);
ellipse(180, 235, 18, 30);
ellipse(220, 235, 18, 30);
triangle(200, 275, 200, 250, 225, 275);
```

C. Eyes 3






```
//Draw the outer circle
fill(255);
ellipse(180, 230, 20, 20);
ellipse(220, 230, 20, 20);
//draw the pupil
fill(0);
ellipse(180, 230, 5, 5);
ellipse(220, 230, 5, 5);
```

D. Other

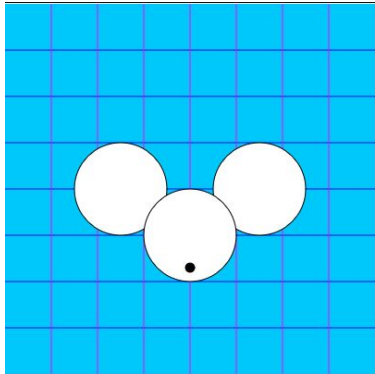
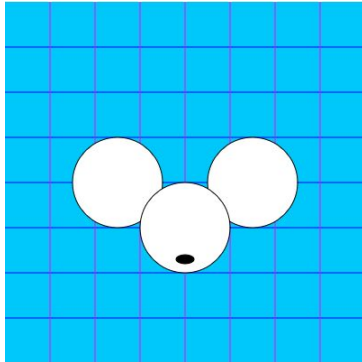
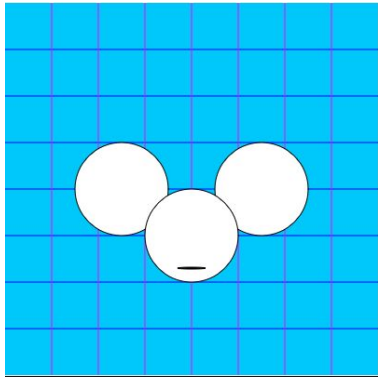
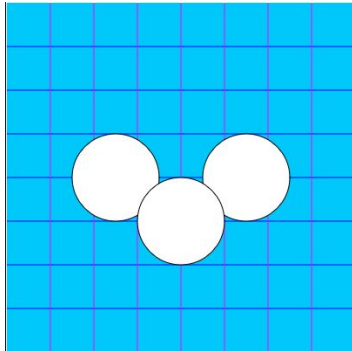


```
//Draw the outer circle
fill(__);
ellipse(__, __, __, __);
ellipse(__, __, __, __);
//draw the pupil
fill(0);
ellipse(__, __, __, __);
ellipse(__, __, __, __);
```

➤ Select a color for the mouth or create your own!

A.  <code>fill(0, 0, 0); //black</code>	B.  <code>fill(255, 0, 0); //red</code>
C.  <code>fill(57, 255, 20); //light green</code>	D. <code>fill(____,____,____); //other</code>

➤ Finally, select a mouth or create your own!

<p>A. Mouth 1</p>  <p><code>ellipse(200, 285, 10, 10);</code></p>	<p>B. Mouth 2</p>  <p><code>ellipse(200, 285, 20, 10);</code></p>
<p>C. Mouth 3</p>  <p><code>ellipse(200, 285, 30, 2);</code></p>	<p>D. Other</p>  <p><code>ellipse(____, ____, ____, ____);</code></p>

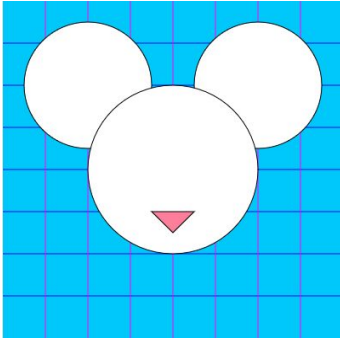
➤ When these steps are finished, turn in your worksheet!

B. Head 2 Options

For each of the following, circle the option you picked. If you create your own, fill in the command with the numbers you used!

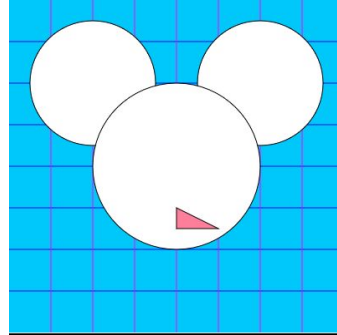
➤ Pick a nose from the following options, or create your own.

A. Nose 1



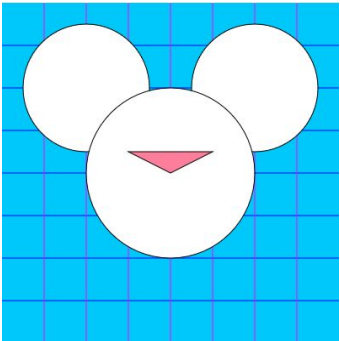
`triangle(200, 275, 175, 250, 225, 250);`

B. Nose 2



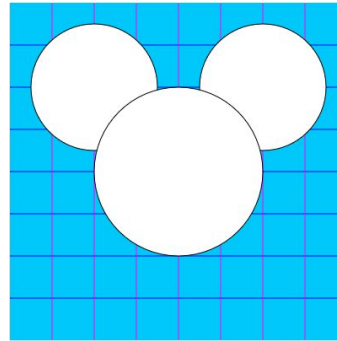
`triangle(200, 275, 200, 250, 250, 275);`

C. Nose 3



`triangle(200, 200, 150, 175, 250, 175);`

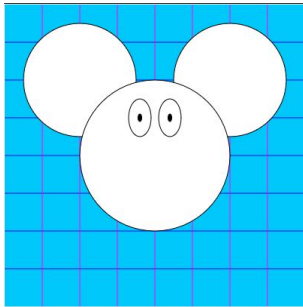
D. Other



`triangle(____, ____, ____, ____, ____, ____);`

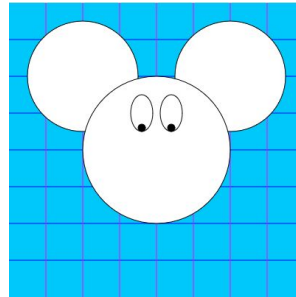
➤ Now we will draw the eyes. Circle one of the following options, or create your own.

A. Eyes 1



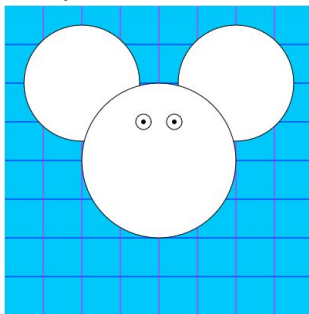
```
//Draw the outer circle
fill(255);
ellipse(180, 150, 30, 50);
ellipse(220, 150, 30, 50);
//draw the pupil
fill(0);
ellipse(180, 150, 5, 10);
ellipse(220, 150, 5, 10);
```

B. Eyes 2



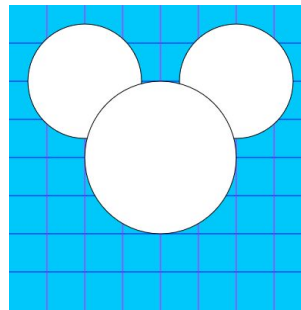
```
//Draw the outer circle
fill(255);
ellipse(180, 150, 30, 50);
ellipse(220, 150, 30, 50);
//draw the pupil
fill(0);
ellipse(180, 170, 10, 10);
ellipse(220, 170, 10, 10);
```

C. Eyes 3






```
//Draw the outer circle
fill(255);
ellipse(180, 150, 20, 20);
ellipse(220, 150, 20, 20);
//draw the pupil
fill(0);
ellipse(180, 150, 5, 5);
ellipse(220, 150, 5, 5);
```

D. Other

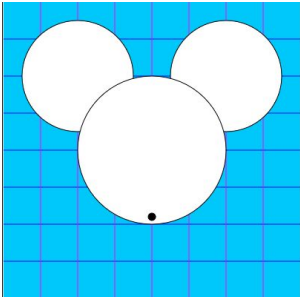
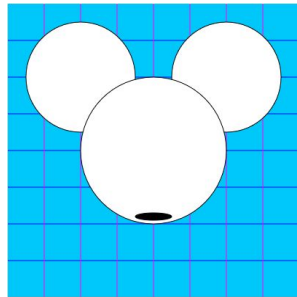
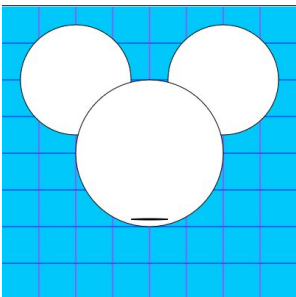
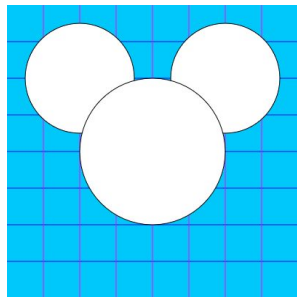


```
//Draw the outer circle
fill(__);
ellipse(__, __, __, __);
ellipse(__, __, __, __);
//draw the pupil
fill(0);
ellipse(__, __, __, __);
ellipse(__, __, __, __);
```

➤ Select a color for the mouth or create your own!

A.  <code>fill(0, 0, 0); //black</code>	B.  <code>fill(255, 0, 0); //red</code>
C.  <code>fill(57, 255, 20); //light green</code>	D. <code>fill(____,____,____); //other</code>

➤ Finally, select a mouth or create your own!

<p>A. Mouth 1</p>  <p><code>ellipse(200, 290, 10, 10);</code></p>	<p>B. Mouth 2</p>  <p><code>ellipse(200, 290, 50, 10);</code></p>
<p>C. Mouth 3</p>  <p><code>ellipse(200, 290, 50, 2);</code></p>	<p>D. Other</p>  <p><code>ellipse(____,____,____,____);</code></p>

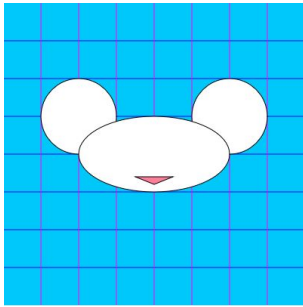
➤ When these steps are finished, turn in your worksheet!

C. Head 3 Options

For each of the following, circle the option you picked. If you create your own, fill in the command with the numbers you used!

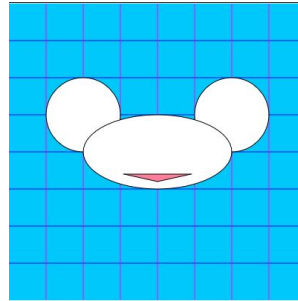
➤ Pick a nose from the following options, or create your own.

A. Nose 1



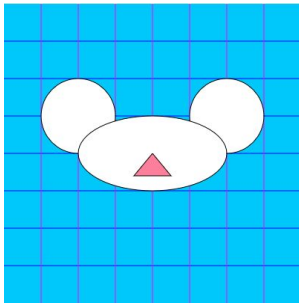
`triangle(200, 240, 175, 230, 225, 230);`

B. Nose 2



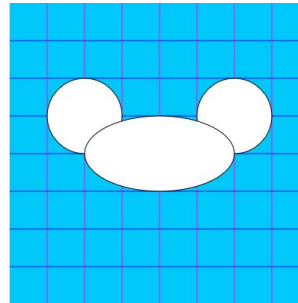
`triangle(200, 240, 155, 230, 245, 230);`

C. Nose 3



`triangle(200, 200, 175, 230, 225, 230);`

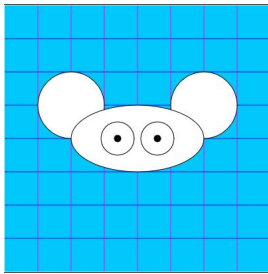
D. Other



`triangle(____, ____, ____, ____, ____, ____);`

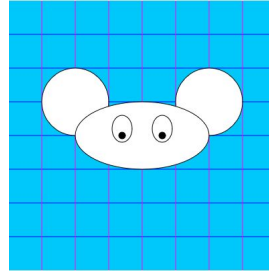
➤ Now we will draw the eyes. Circle one of the following options, or create your own.

A. Eyes 1



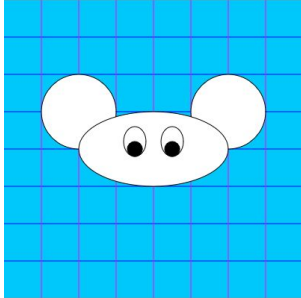
```
//Draw the outer circle
fill(255);
ellipse(170, 200, 50, 50);
ellipse(230, 200, 50, 50);
//draw the pupil
fill(0);
ellipse(170, 200, 10, 10);
ellipse(230, 200, 10, 10);
```

B. Eyes 2



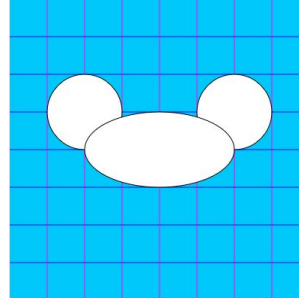
```
//Draw the outer circle
fill(255);
ellipse(170, 190, 30, 40);
ellipse(230, 190, 30, 40);
//draw the pupil
fill(0);
ellipse(170, 200, 10, 10);
ellipse(230, 200, 10, 10);
```

C. Eyes 3






```
//Draw the outer circle
fill(255);
ellipse(175, 190, 30, 40);
ellipse(225, 190, 30, 40);
//draw the pupil
fill(0);
ellipse(175, 200, 20, 20);
ellipse(225, 200, 20, 20);
```

D. Other

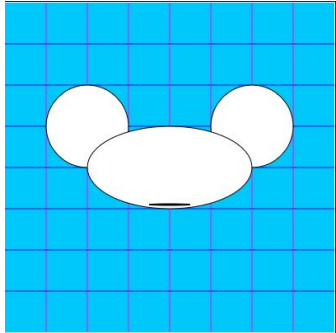
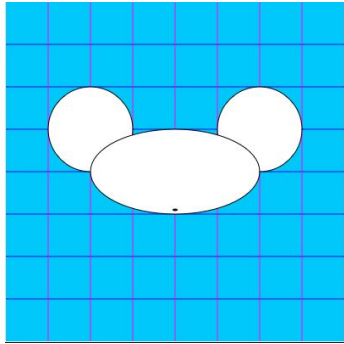
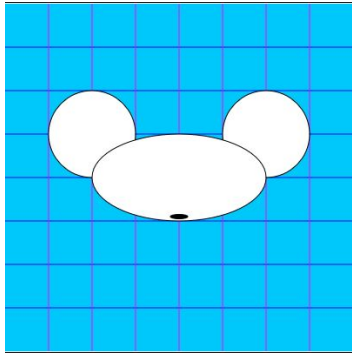
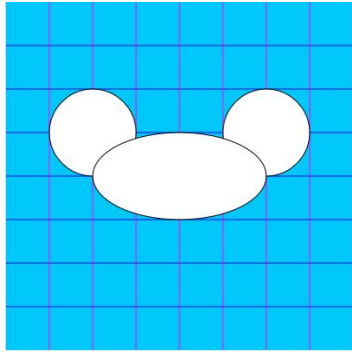


```
//Draw the outer circle
fill(__);
ellipse(__, __, __, __);
ellipse(__, __, __, __);
//draw the pupil
fill(0);
ellipse(__, __, __, __);
ellipse(__, __, __, __);
```

➤ Select a color for the mouth or create your own!

A.  <code>fill(0, 0, 0); //black</code>	B.  <code>fill(255, 0, 0); //red</code>
C.  <code>fill(57, 255, 20); //light green</code>	D. <code>fill(____,____,____); //other</code>

➤ Finally, select a mouth or create your own!

<p>A. Mouth 1</p>  <p><code>ellipse(200, 245, 50, 2);</code></p>	<p>B. Mouth 2</p>  <p><code>ellipse(200, 245, 5, 2);</code></p>
<p>C. Mouth 3</p>  <p><code>ellipse(200, 245, 20, 5);</code></p>	<p>D. Other</p>  <p><code>ellipse(____,____,____,____);</code></p>

➤ When these steps are finished, turn in your worksheet!

D. Head 4 Options

Use the outline to create your own mouse. There is a grid on the next page if you would like to draw it to help you.

➤ First, create a triangle nose

```
triangle(____, ____, ____, ____, ____, ____);
```

➤ Now we will draw the eyes with four ellipses - two for the outside and two for the pupils.

```
//Draw the outer circle
```

```
fill(____);
```

```
ellipse(____,____,____,____);
```

```
ellipse(____,____,____,____);
```

```
//draw the pupil
```

```
fill(0);
```

```
ellipse(____,____,____,____);
```

```
ellipse(____,____,____,____);
```

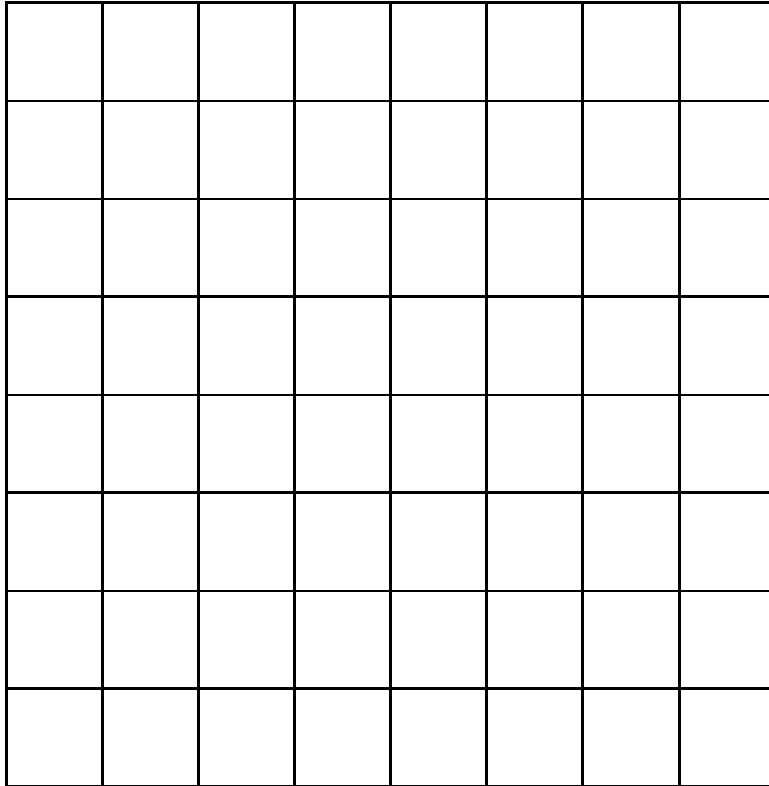
➤ Next, select a color for the mouth and draw an ellipse for the mouth.

```
fill(____,____,____);
```

```
ellipse(____,____,____,____);
```

➤ When these steps are finished, turn in your worksheet!

$(0,0)$



$(400, 400)$

Appendix D

PEABODY-INTERVENTION 1: CC SNOWMAN

The following is the curriculum for intervention 1 (testing student preferences concerning creative choice while practicing concepts). This is the set of worksheets where students could use creative choice to create a snowman.

Practice Problems-Processing Basics

J. Wilcox, Z. Wood, K. Mork

Today we are going to do some practice by creating a snowman! This week, choose which options you like best or create your own options as you follow the steps.

➤ First, let's copy the grid code to help draw our snowman.

Google search: **Zoe Wood**

Click on: **Base Code for grid**




Click on the page and type the command to select all: **cnt-a**

Type the command to copy: **cnt-c**

Go to the processing editor and click on the screen and type the command to paste: **cnt-v**

➤ **For each of the following, circle the option you pick** - if you create your own be sure to **write the numbers in for the command you chose.**




➤ Next, let's pick a background color to place our snowman in. Pick from the following or make up your own. Find the line "background(255);" in the code you pasted and **replace it with your choice.**

A. Light Blue  background(0, 200, 250);	B. Dark Blue  background(17, 30, 108);
C. Light Purple  background(150, 111, 214);	D. Other background(____,____,____);

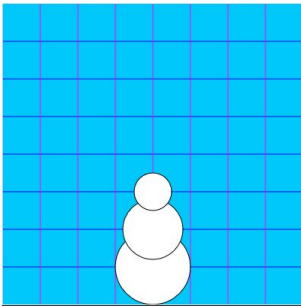
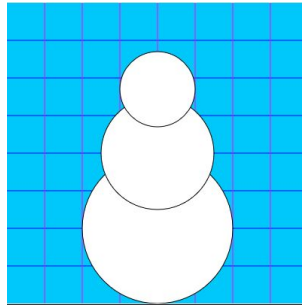
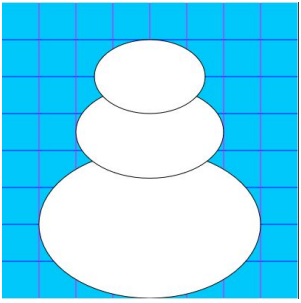
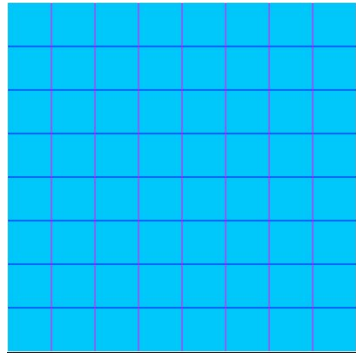
➤ Now let's pick an outline color for our snowman. Type one of the following:

A. stroke(0); //black	B. stroke(0, 255, 0); //green
C. noStroke(); //no outline	D. stroke(____,____,____); //other




➤ Next, we will create the snowman body. Circle the color you want or write your own:

A.  fill(255); //white	B.  fill(50); //gray
C.  fill(139, 25, 155); //dark purple	D. fill(____,____,____); //other

➤ Select one of the following or create your own snowman for option D. You can use the grid on the next page to help you design a snowman if you would like. If you choose option D, you will have to keep designing the rest too since the example features might not fit the size you choose. Circle the option you chose and type the corresponding code:

<p>A. Body 1</p>  <pre>ellipse(200, 350, 100, 100); ellipse(200, 300, 80, 80); ellipse(200, 250, 50, 50);</pre>	<p>B. Body 2</p>  <pre>ellipse(200, 300, 200, 200); ellipse(200, 200, 150, 150); ellipse(200, 115, 100, 100);</pre>
<p>C. Body 3</p>  <pre>ellipse(200, 300, 300, 200); ellipse(200, 175, 200, 125); ellipse(200, 100, 150, 100);</pre>	<p>D. Other</p>  <pre>ellipse(____, ____, ____, ____); ellipse(____, ____, ____, ____); ellipse(____, ____, ____, ____);</pre>

➤ Select a color for the nose!

A.  fill(254, 127, 156); //pink	B.  fill(249, 166, 2); //orange
C.  fill(57, 255, 20); //light green	D. fill(____,____,____); //other

➤ Now, go to the option for the head you picked.

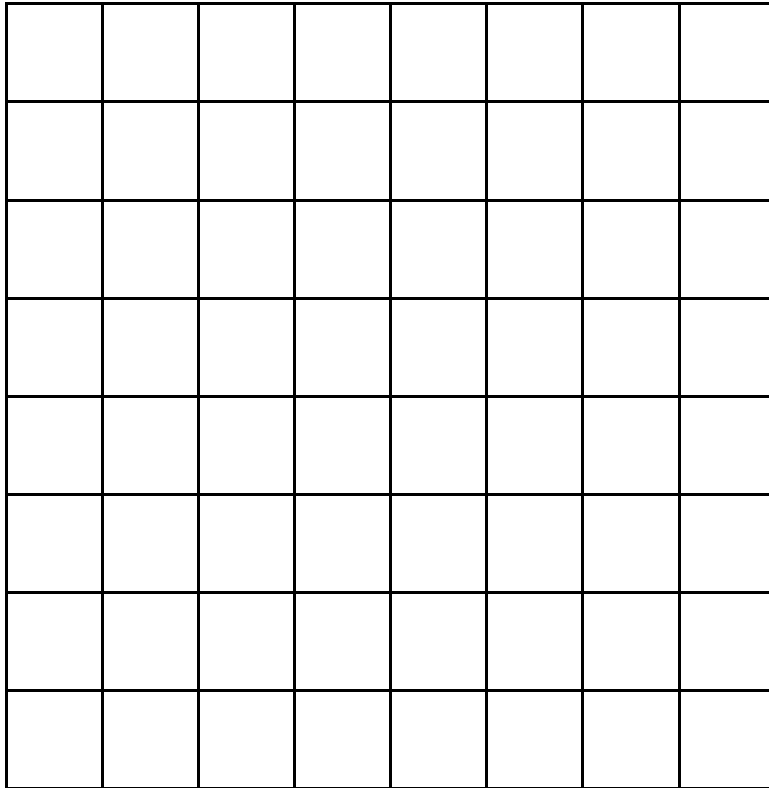
A: go to page 4

B: go to page 7

C: go to page 10

D: go to page 13

(0,0)



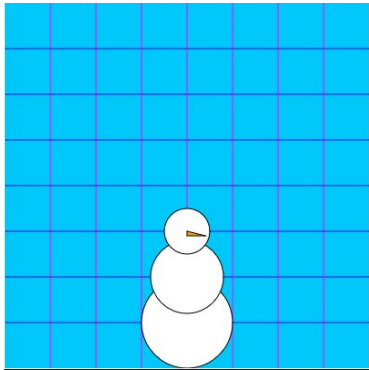
(400, 400)

A. Body 1 Options

For each of the following, circle the option you picked. If you create your own, fill in the command with the numbers you used!

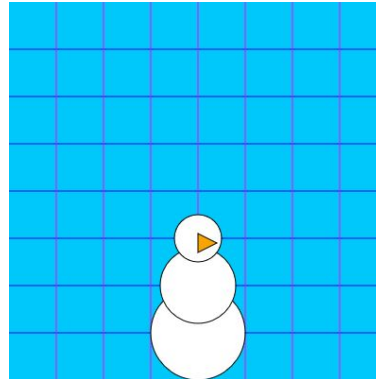
➤ Pick a nose from the following options, or create your own.

A. Nose 1



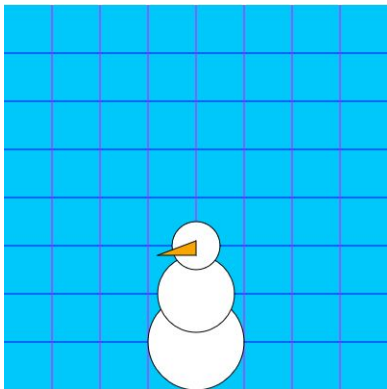
triangle(200, 250, 200, 255, 220, 255);

B. Nose 2



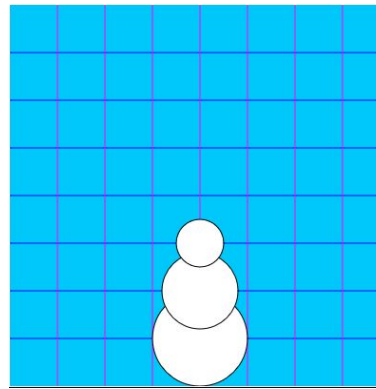
triangle(200, 245, 200, 265, 220, 255);

C. Nose 3



triangle(200, 245, 200, 260, 160, 260);

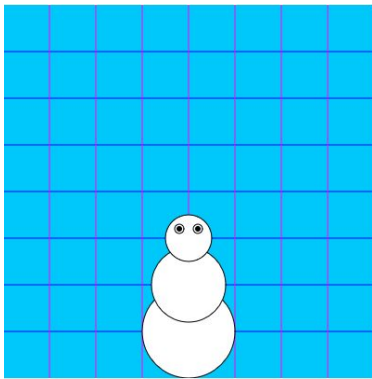
D. Other



triangle(____, ____, ____, ____, ____, ____);

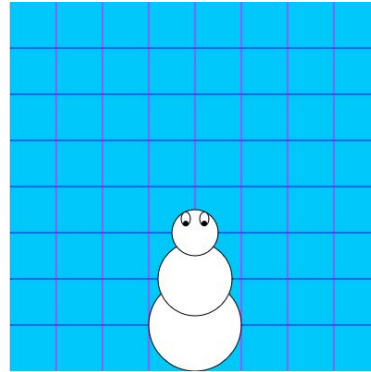
➤ Now we will draw the eyes. Circle one of the following options, or create your own.

A. Eyes 1



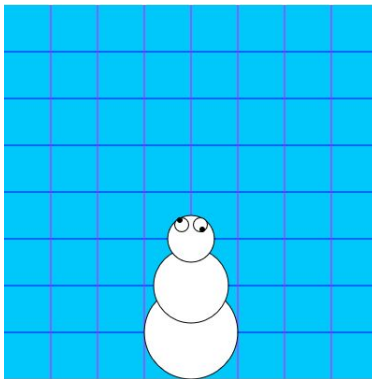
```
//Draw the outer circle
fill(255);
ellipse(190, 240, 10, 10);
ellipse(210, 240, 10, 10);
//draw the pupil
fill(0);
ellipse(190, 240, 5, 5);
ellipse(210, 240, 5, 5);
```

B. Eyes 2



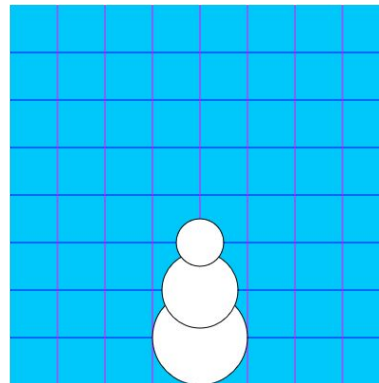
```
//Draw the outer circle
fill(255);
ellipse(190, 235, 10, 15);
ellipse(210, 235, 10, 15);
//draw the pupil
fill(0);
ellipse(190, 240, 5, 5);
ellipse(210, 240, 5, 5);
```

C. Eyes 3






```
//Draw the outer circle
fill(255);
ellipse(190, 235, 15, 15);
ellipse(210, 235, 15, 15);
//draw the pupil
fill(0);
ellipse(188, 230, 5, 5);
ellipse(212, 240, 5, 5);
```

D. Other

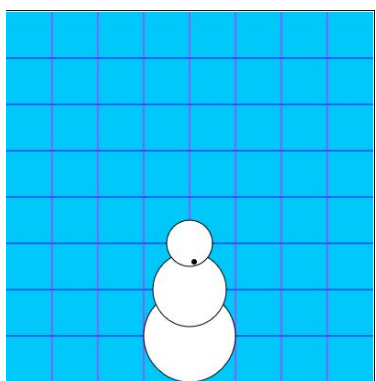
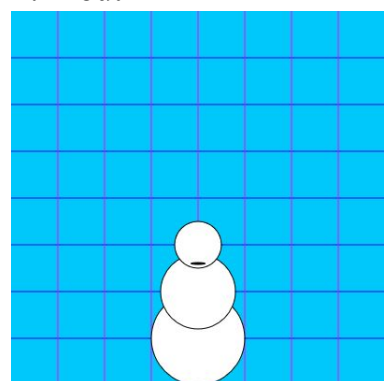
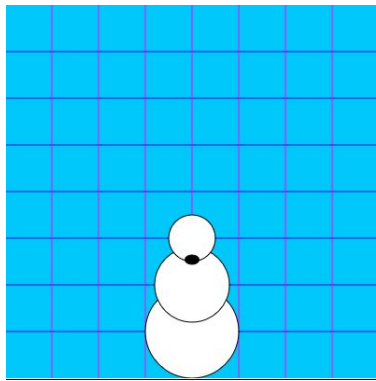
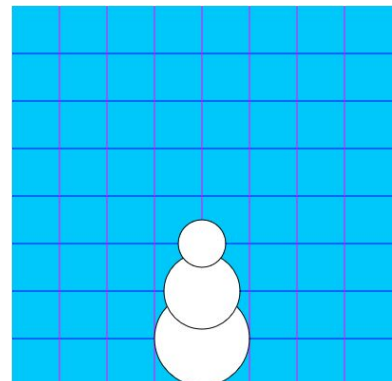


```
//Draw the outer circle
fill(__);
ellipse(__, __, __, __);
ellipse(__, __, __, __);
//draw the pupil
fill(0);
ellipse(__, __, __, __);
ellipse(__, __, __, __);
```


➤ Select a color for the mouth or create your own!

A.  <code>fill(0, 0, 0); //black</code>	B.  <code>fill(255, 0, 0); //red</code>
C.  <code>fill(57, 255, 20); //light green</code>	D. <code>fill(____,____,____); //other</code>

➤ Finally, select a mouth or create your own!

<p>A. Mouth 1</p>  <p><code>ellipse(205, 270, 5, 5);</code></p>	<p>B. Mouth 2</p>  <p><code>ellipse(200, 270, 15, 2);</code></p>
<p>C. Mouth 3</p>  <p><code>ellipse(200, 273, 15, 10);</code></p>	<p>D. Other</p>  <p><code>ellipse(____, ____, ____, ____);</code></p>

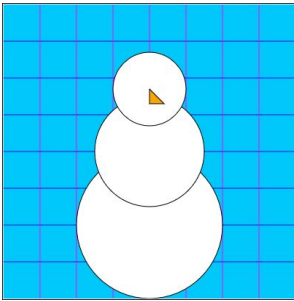
➤ When these steps are finished, turn in your worksheet and you can go online to fill out the survey.

B. Body 2 Options

For each of the following, circle the option you picked. If you create your own, fill in the command with the numbers you used!

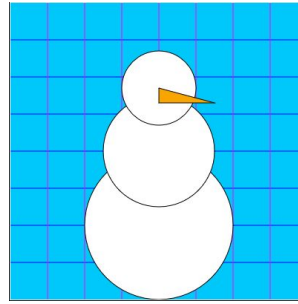
➤ Pick a nose from the following options, or create your own.

A. Nose 1



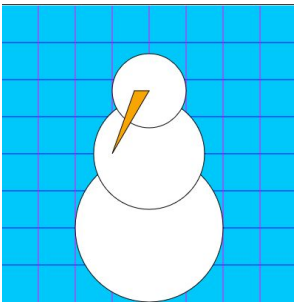
`triangle(200, 115, 200, 135, 220, 135);`

B. Nose 2



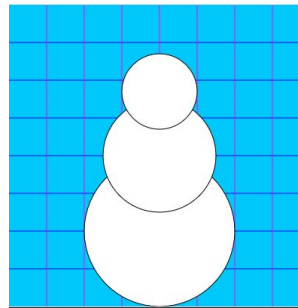
`triangle(200, 115, 200, 135, 275, 135);`

C. Nose 3



`triangle(200, 115, 150, 200, 180, 115);`

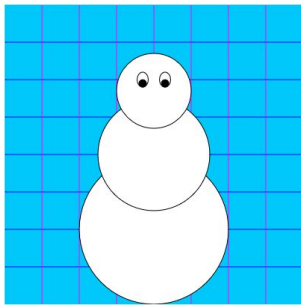
D. Other



`triangle(____, ____, ____, ____, ____, ____);`

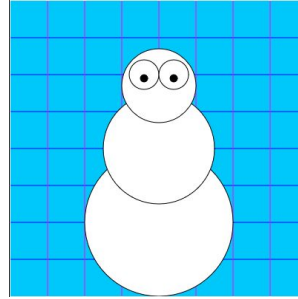
➤ Now we will draw the eyes. Circle one of the following options, or create your own.

A. Eyes 1



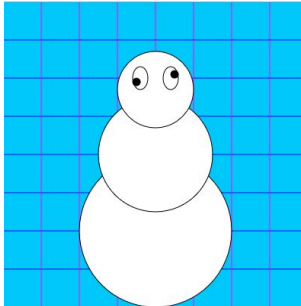
```
//Draw the outer circle
fill(255);
ellipse(185, 100, 15, 20);
ellipse(215, 100, 15, 20);
//draw the pupil
fill(0);
ellipse(185, 105, 10, 10);
ellipse(215, 105, 10, 10);
```

B. Eyes 2



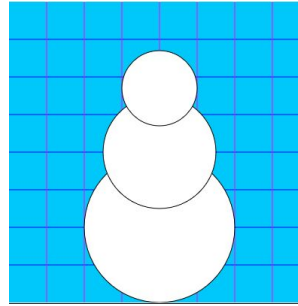
```
//Draw the outer circle
fill(255);
ellipse(180, 100, 40, 40);
ellipse(220, 100, 40, 40);
//draw the pupil
fill(0);
ellipse(180, 105, 10, 10);
ellipse(220, 105, 10, 10);
```

C. Eyes 3






```
//Draw the outer circle
fill(255);
ellipse(180, 100, 20, 30);
ellipse(220, 100, 20, 30);
//draw the pupil
fill(0);
ellipse(175, 105, 10, 10);
ellipse(225, 95, 10, 10);
```

D. Other

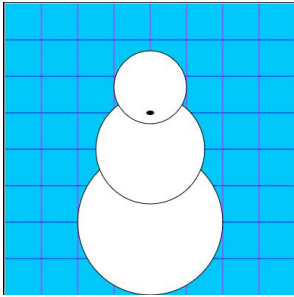
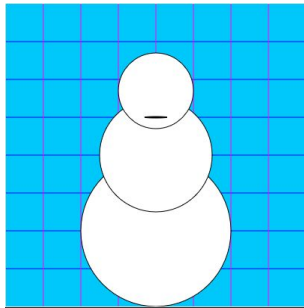
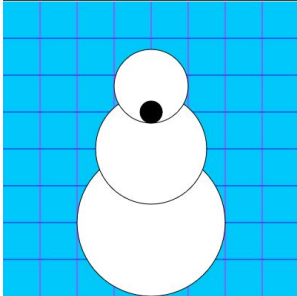
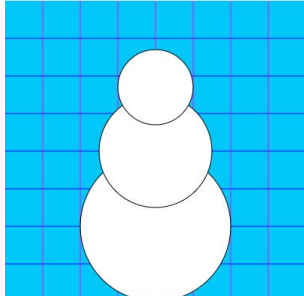


```
//Draw the outer circle
fill(__);
ellipse(__, __, __, __);
ellipse(__, __, __, __);
//draw the pupil
fill(0);
ellipse(__, __, __, __);
ellipse(__, __, __, __);
```

➤ Select a color for the mouth or create your own!

A.  <code>fill(0, 0, 0); //black</code>	B.  <code>fill(255, 0, 0); //red</code>
C.  <code>fill(57, 255, 20); //light green</code>	D. <code>fill(____,____,____); //other</code>

➤ Finally, select a mouth or create your own!

<p>A. Mouth 1</p>  <p><code>ellipse(200, 150, 10, 5);</code></p>	<p>B. Mouth 2</p>  <p><code>ellipse(200, 150, 30, 2);</code></p>
<p>C. Mouth 3</p>  <p><code>ellipse(200, 150, 30, 30);</code></p>	<p>D. Other</p>  <p><code>ellipse(____, ____, ____, ____);</code></p>

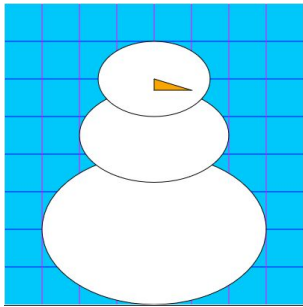
➤ When these steps are finished, turn in your worksheet and you can go online to fill out the survey.

C. Body 3 Options

For each of the following, circle the option you picked. If you create your own, fill in the command with the numbers you used!

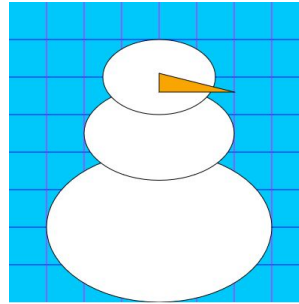
➤ Pick a nose from the following options, or create your own.

A. Nose 1



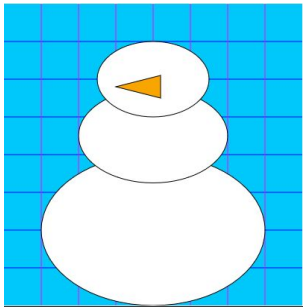
`triangle(200, 100, 200, 115, 250, 115);`

B. Nose 2



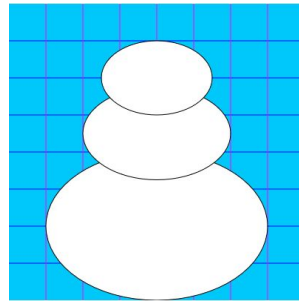
`triangle(200, 95, 200, 120, 300, 120);`

C. Nose 3



`triangle(210, 95, 210, 125, 150, 110);`

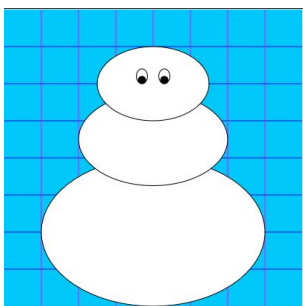
D. Other



`triangle(____, ____, ____, ____, ____, ____);`

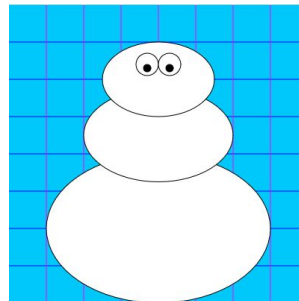
➤ Now we will draw the eyes. Circle one of the following options, or create your own.

A. Eyes 1

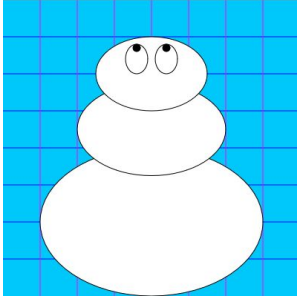
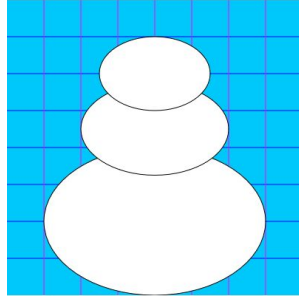


`//Draw the outer circle`




B. Eyes 2



`//Draw the outer circle
fill(255);`

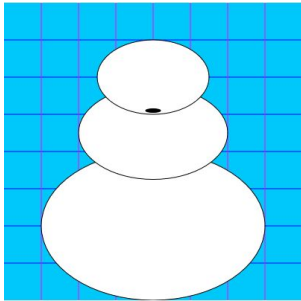
<pre>fill(255); ellipse(185, 90, 15, 20); ellipse(215, 90, 15, 20); //draw the pupil fill(0); ellipse(185, 95, 10, 10); ellipse(215, 95, 10, 10);</pre>	<pre>ellipse(185, 80, 30, 30); ellipse(215, 80, 30, 30); //draw the pupil fill(0); ellipse(185, 85, 10, 10); ellipse(215, 85, 10, 10);</pre>
<p>C. Eyes 3</p>  <pre>//Draw the outer circle fill(255); ellipse(180, 80, 30, 40); ellipse(220, 80, 30, 40); //draw the pupil fill(0); ellipse(180, 65, 10, 10); ellipse(220, 65, 10, 10);</pre>	<p>D. Other</p>  <pre>//Draw the outer circle fill(__); ellipse(__, __, __, __); ellipse(__, __, __, __); //draw the pupil fill(0); ellipse(__, __, __, __); ellipse(__, __, __, __);</pre>

➤ Select a color for the mouth or create your own!

<p>A.  fill(0, 0, 0); //black</p>	<p>B.  fill(255, 0, 0); //red</p>
<p>C.  fill(57, 255, 20); //light green</p>	<p>D. fill(__, __, __); //other</p>

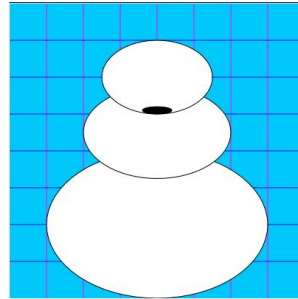
➤ Finally, select a mouth or create your own!

A. Mouth 1



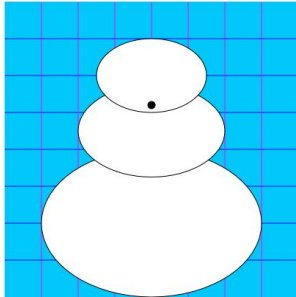
`ellipse(200, 145, 20, 5);`

B. Mouth 2



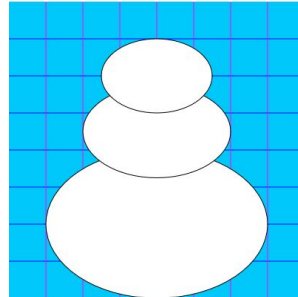
`ellipse(200, 145, 40, 10);`

C. Mouth 3



`ellipse(200, 140, 10, 10);`

D. Other



`ellipse(____, ____, ____, ____);`

➤ When these steps are finished, turn in your worksheet and you can go online to fill out the survey.

D. Body 4 Options

Use the outline to create your own snowman.

➤ First, create a triangle nose

```
triangle(__, __, __, __, __, __);
```

➤ Now we will draw the eyes with four ellipses - two for the outside and two for the pupils.

```
//Draw the outer circle
```

```
fill(__);
```

```
ellipse(__, __, __, __);
```

```
ellipse(__, __, __, __);
```

```
//draw the pupil
```

```
fill(0);
```

```
ellipse(__, __, __, __);
```

```
ellipse(__, __, __, __);
```

➤ Next, select a color for the mouth and draw an ellipse for the mouth.

```
fill(__, __, __);
```

```
ellipse(__, __, __, __);
```

➤ When these steps are finished, turn in your worksheet and you can go online to fill out the survey.

Appendix E

PEABODY-INTERVENTION 1: NO CC MOUSE

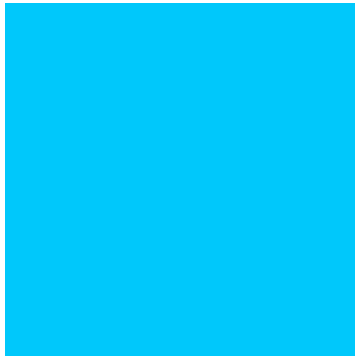
The following is the curriculum for intervention 1 (testing student preferences concerning creative choice while practicing concepts). This is the set of worksheets where students could not use creative choice to create a mouse.

Practice Problems-Processing Basics

J. Wilcox, Z. Wood, K. Mork

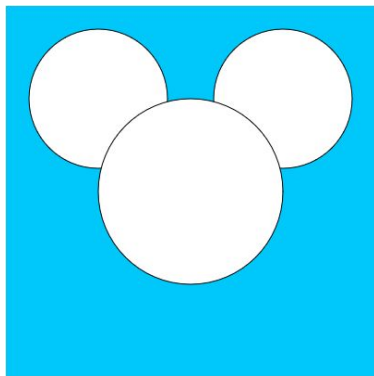
Today we are going to do some practice by creating a mouse! This week, just follow the instructions exactly as given.

➤ First, let's make the canvas. Fill in the size and blue color by typing the following commands:



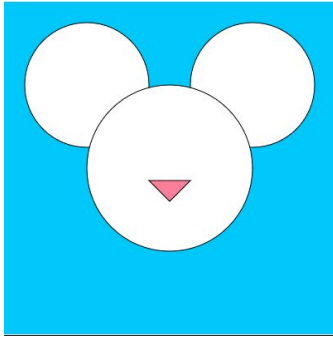
```
size(400, 400);  
background(0, 200, 250);
```

➤ Next, we will create the mouse head. Type the following command to add the three ellipses:



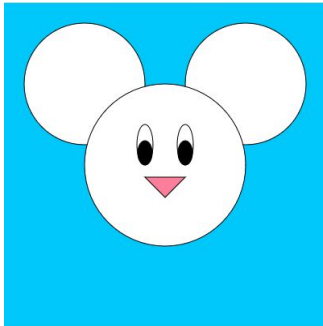
```
ellipse(100, 100, 150, 150);  
ellipse(300, 100, 150, 150);  
ellipse(200, 200, 200, 200);
```

➤ Select pink and draw the triangle nose!



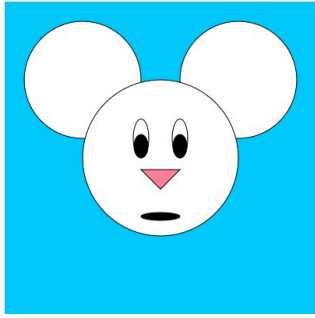
```
fill(254, 127, 156);  
triangle(200, 240, 175, 215, 225, 215);
```

➤ Next, we will add eyes! This will take four ellipses - two white and two black.



```
//Draw the outside of the eyes  
fill(255);  
ellipse(175, 175, 20, 50);  
ellipse(225, 175, 20, 50);  
//draw the pupil  
fill(0);  
ellipse(175, 185, 18, 30);  
ellipse(225, 185, 18, 30);
```

➤ Finally, we will add a mouth.



```
ellipse(200, 275, 50, 10);
```

➤ When these steps are finished, raise your hand for us to check your assignment!

Appendix F

PEABODY-INTERVENTION 1: NO CC SNOWMAN

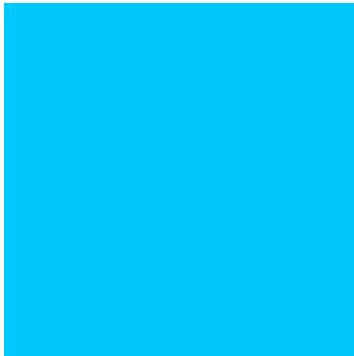
The following is the curriculum for intervention 1 (testing student preferences concerning creative choice while practicing concepts). This is the set of worksheets where students could not use creative choice to create a snowman.

Practice Problems-Processing Basics

J. Wilcox, Z. Wood, K. Mork

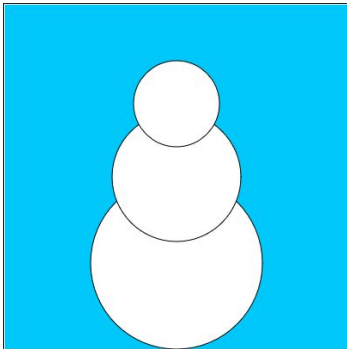
Today we are going to do some practice by creating a snowman! This week, just follow the instructions exactly as given.

➤ First, let's make the canvas. Fill in the size and blue color by typing the following commands:



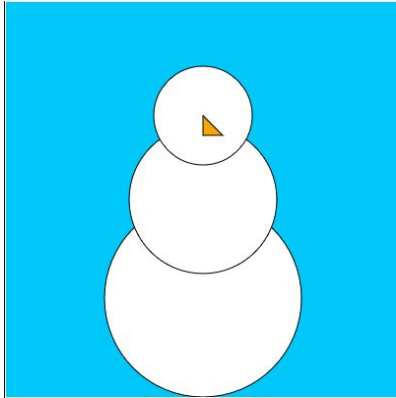
```
size(400, 400);  
background(0, 200, 250);
```

➤ Next, we will create the snowman. Type the following command to add the three ellipses:



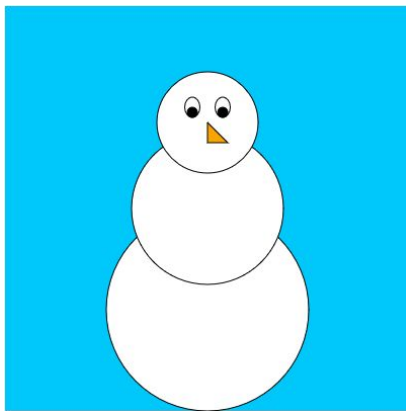
```
ellipse(200, 300, 200, 200);  
ellipse(200, 200, 150, 150);  
ellipse(200, 115, 100, 100);
```

➤ Finally, select orange and draw the triangle nose!



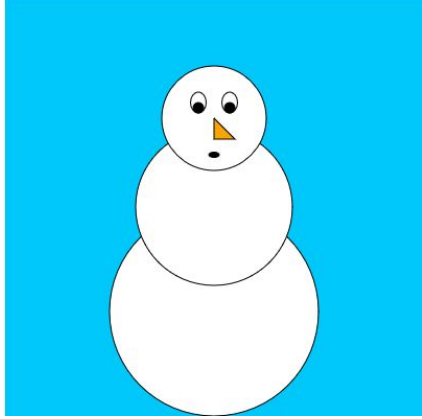
```
fill(253, 106, 2);  
triangle(200, 115, 200, 135, 220, 135);
```

➤ Next, we will add eyes! This will take four ellipses - two white and two black.



```
//Draw the outer circle  
fill(255);  
ellipse(185, 100, 15, 20);  
ellipse(215, 100, 15, 20);  
//draw the pupil  
fill(0);  
ellipse(185, 105, 10, 10);  
ellipse(215, 105, 10, 10);
```


➤ Finally, we will add a mouth.



```
ellipse(200, 150, 10, 5);
```

➤ When these steps are finished, raise your hand for us to check your assignment!

Appendix G

PEABODY-INTERVENTION 1: POST PRACTICE SURVEY

The following is the survey the fifth graders were given after two weeks of worksheets were completed (one week with creative choice and one week without).

Post-Practice Survey

Answer the following questions honestly - your answers will stay anonymous!

* Required

1. Enter Your lunch ID: *

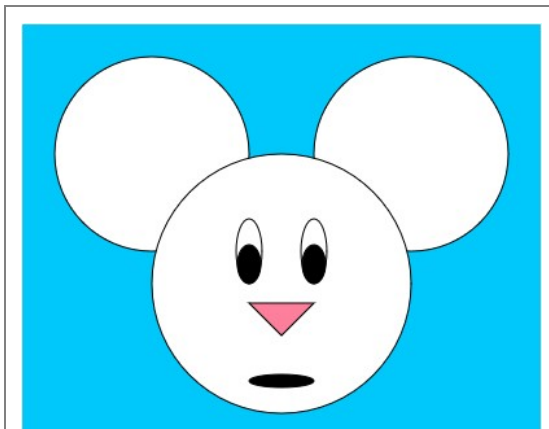
2. Select the Lab Period *

Mark only one oval.

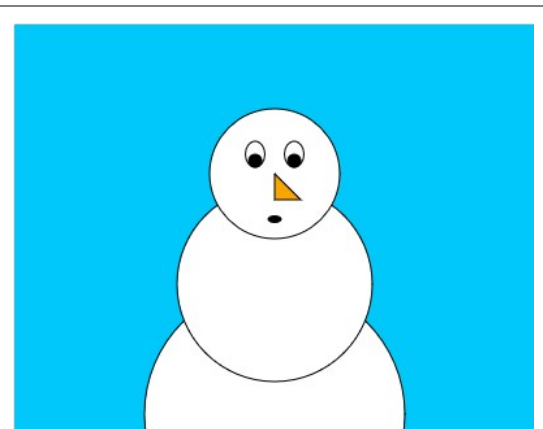
- ☐ Monday 12:45
- ☐ Monday 1:55
- ☐ Thursday 8:30
- ☐ Thursday 9:50

3. Think about the recent labs from this week and last week (making a mouse and making a snowman). Which one did you like better?

Mark only one oval.



☐ making a mouse



☐ making a snowman

4. Why did you like this option better?*Check all that apply.*

- ☐ it was more creative
- ☐ mice are cuter
- ☐ more of a fun challenge
- ☐ it had easier instructions
- ☐ I was able to finish it
- ☐ snowmen are cooler
- ☐ more freedom
- ☐ I understood it better
- ☐ Other: _____

5. Please write the most important of the box you checked:

6. One week you were told to follow instructions, and one of the weeks you could pick the different colors and shapes. Which did you like better?*Mark only one oval.*

- ☐ I liked it better when we followed instructions
- ☐ I liked it better when we picked from options/made our own options

7. Why did you like this option better?

Powered by



Appendix H

PEABODY-INTERVENTION 2: CC CONDITIONALS

The following is the curriculum for intervention 2 (testing student preferences concerning creative choice while learning a new concepts, specifically conditionals). This is the set of worksheets (part 1 and part 2) where students were given creative choice while working with this new concept.

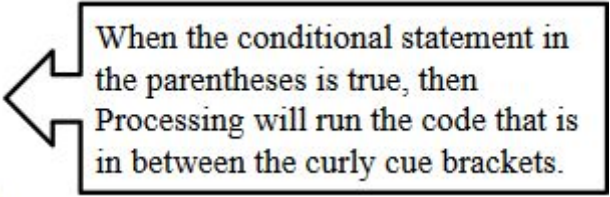
Conditionals Part 1

J. Wilcox, Z. Wood, K. Mork

Introduction: Changing Background Color

A powerful tool we can learn to use in Processing uses a type of statement called a _____. These statements can either be _____ or _____. The way we enter one of these statements into Processing looks like this:

```
if ( _____ ) {
    'This is where we
    Would write the code'
}
```



When the conditional statement in the parentheses is true, then Processing will run the code that is in between the curly cue brackets.

The conditional statement we make will use one of these relational operators:

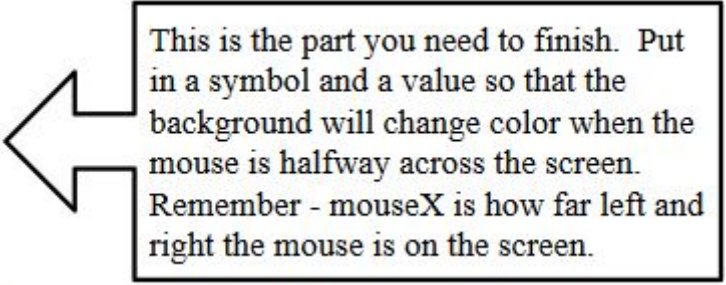
Symbol	Meaning
>	
<	

Finish the following code and type it in.

```
void setup() {
    size(400, 400);
}

void draw() {
    background(50);

    if (mouseX > _____) {
        background(100, 0, 250);
    }
}
```



This is the part you need to finish. Put in a symbol and a value so that the background will change color when the mouse is halfway across the screen. Remember - mouseX is how far left and right the mouse is on the screen.

Now let's try changing more than just background color!

Changing Basic Shapes

Next let's change the shapes that appear on the screen. First, take a look at the following code.

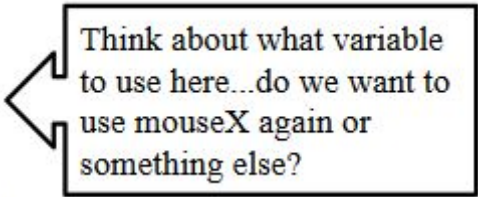
```
void setup() {  
    size(400, 400);  
}  
  
void draw() {  
    background(50);  
    fill(255, 0, 0);  
    ellipse(200, 200, 100, 100);  
}
```

It draws a _____.

See if you can change it to draw a green rectangle when the mouse is on the **top** half of the screen, and draw what it currently draws if the mouse is on the bottom of the screen.

Fill in the following code!

```
void setup() {  
    size(400, 400);  
}  
  
void draw() {  
    background(50);  
    fill(255, 0, 0);  
    ellipse(200, 200, 100, 100);  
  
    if ( _____ ) {  
        fill(0, 255, 0);  
        rect(175, 175, 50, 50);  
    }  
}
```



Think about what variable to use here...do we want to use mouseX again or something else?

What happens now?

Is this what we want to happen? Let's add one more conditional.

```
void setup() {
    size(400, 400);
}

void draw() {
    background(50);

    //draw red ellipse if mouse is
    // on the bottom half of screen
    if (_____ < _____) {
        fill(255, 0, 0);
        ellipse(200, 200, 100, 100);
    }

    //draw a green rectangle if mouse is
    // on the top half of screen
    if (_____ > _____) {
        fill(0, 255, 0);
        rect(175, 175, 50, 50);
    }
}
```

Now let's do a challenge on our own. Switching between a rectangle and ellipse is cool, but what if we did something cooler?

Think of what drawings might be fun to switch between - maybe you want to flip between something similar: a bored face to a scared face, or a dog to a cat, or Minnie Mouse to Mickey Mouse. Or maybe you want to flip between two completely different things!

Changing Drawings

Design two drawings and redo the code above with your new objects in place of the rectangle and ellipse. Fill in the code below and see what you can make!

```
void setup() {  
    size(400, 400);  
}  
  
void draw() {  
    background(50);  
  
    //drawing 1 if mouse is  
    // on the bottom half of screen  
    if (_____ < _____) {  
        _____  
        _____  
        _____  
        _____  
        _____  
        _____  
    }  
  
    //drawing 2 if mouse is  
    // on the top half of screen  
    if (_____ > _____) {  
        _____  
        _____  
        _____  
        _____  
        _____  
        _____  
    }  
}
```

For a final challenge, see if you can add more toggles. We change the drawing if the mouse is on the top or bottom of the screen. Let's also change the background color if the mouse is on the left or right half of the screen.

WITHOUT deleting anything, add the following lines of code inside your draw function. (Note: “inside your draw function” means somewhere in between the curly braces after `void draw()`). Do you want to change the background color BEFORE or AFTER you change the drawing? _____

```
//background color if mouse is on left side of screen
if (mouseX < width/2) {
    background(255, 255, 255);
}

//background color if mouse is on right side of screen
if (mouseX > width/2) {
    background(0, 0, 0);
}
```

Conditionals Part 2

J. Wilcox, Z. Wood, K. Mork

Recap

In our last session we learned about using conditionals in our code. Remember: a conditional statement is something that can be _____ or _____.

A conditional statement can claim something about a variable. What is a variable?

If our variable changes values, our conditional statement might switch from true to false or from false to true. We can change our actions based on if our conditional is true or not.

So far, our conditional statements have looked like this:

variable < some number

variable > some number

The variables we have looked at are mouseX and mouseY.

Let's think about our variable mouseX again. What does mouseX tell us?

Look at the following conditional statement for a canvas of size(800, 400):

```
mouseX < 400;
```

What does this statement claim? (remember that our canvas has a width of 800).

Is the conditional statement true if the mouse is on the left side of the screen? _____

Is the conditional statement true if the mouse is on the right side of the screen? _____

What conditional statement would claim the mouse is on the right side of the screen?

Variable < or > some number

Remember that we can use conditional statements about mouseX to change our picture. What does the following code do?

```
float px = 0;

void setup() {
  size(800, 400);
}

void draw() {
  background(0, 255, 255);

  if (mouseX < 400) {
    fill(255, 255, 0);
    ellipse(px, 200, 50, 50);
  }
  px = px + 1;
}
```

Statements with Other Variables

We can do a lot by making statements about mouseX and mouseY, but what else can we do? Can we also make statements about other variables? Let's think about what other variables we have used!

Look at the code above and see if you can find any other variables. Another variable our code uses, besides mouseX, is _____. This variable is being used to represent _____

Let's make a statement that the x-position of the ellipse is on the right half of a 800 width canvas. How could I write that statement?

_____ < or > _____
Variable some number

Now, instead of seeing if our cursor crosses halfway across the screen, let's see if our ellipse does!

Fill in the code to write: if the ellipse is on the left half of the screen then pick one color, and if the ellipse is on the right half of the screen then pick another color.

```
float px = 0;

void setup() {
    size(800, 400);
}

void draw() {

    //ellipse is on left
    if (_____) {
        background(0, 255, 255);
        fill(255, 255, 0);
    }
    //ellipse is on right
    if (_____) {
        background(0, 0, 0);
        fill(255, 255, 255);
    }

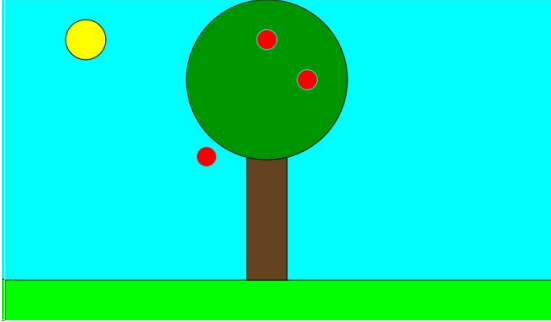
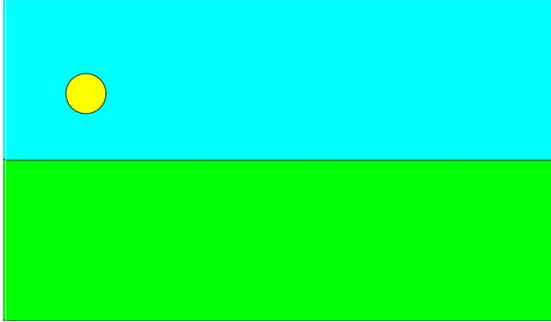
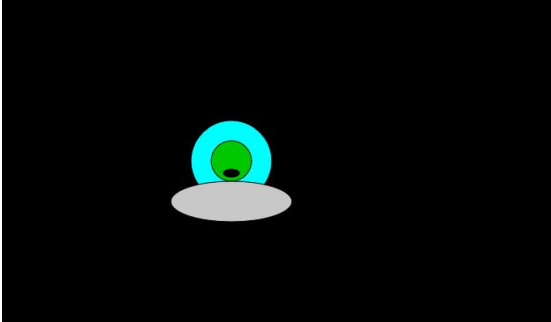
    ellipse(px, 100, 50, 50);
    px = px + 1;
}
```

Challenge Question: What piece of code could you add to make the ellipse go back to the beginning after it passes the end of the canvas?

Practice

Now it's time to create! (You can use the code on the previous page as a starting point if you need). For the rest of this lab, make any scene that has the following.

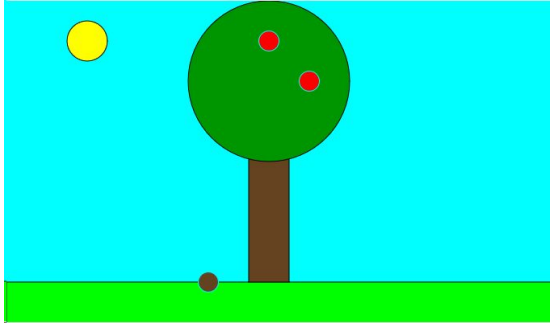
1. Add at least one moving piece (moving any direction). For example:

<p>A. Scene of a fruit tree with one of the apples falling down.</p> 	<p>B. Scene with a sun setting from top of the screen to the bottom</p> 
<p>C. A UFO flying across the sky</p> 	<p>D. Anything else you can think of!</p>

Go to the next page for the next steps!

2. Add at least one conditional statement that changes something based on the moving piece's position

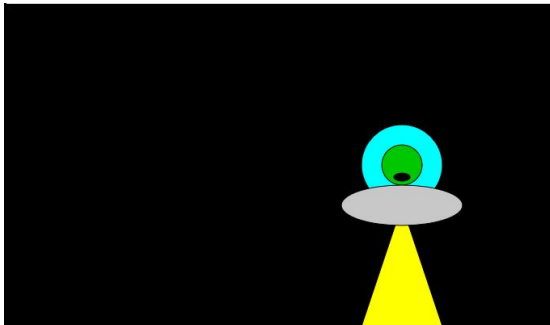
A. When the position of the fruit hits the ground, have it stop moving. (What number will you need to use in the conditional for this?) Maybe you could make the apple shrink or change colors when it hits the ground as well!



B. When the sun crosses to the bottom half of the screen, make the sky and grass darker



C. If the UFO crosses to the right half of the screen, add a tractor beam

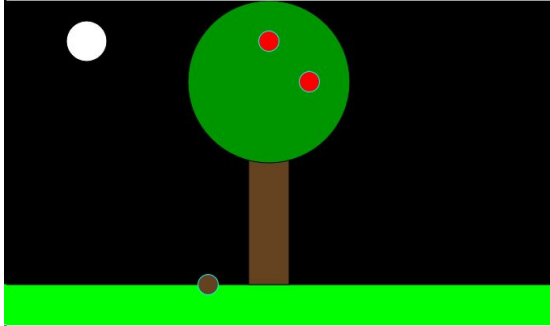


D. Anything else you can think of!

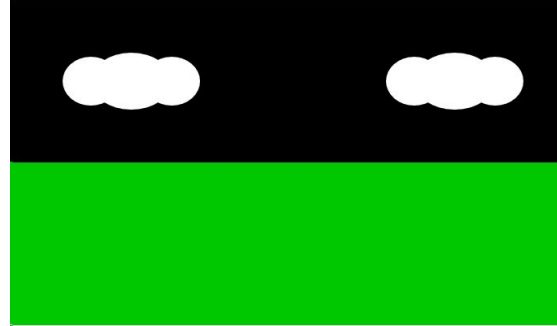
Go to the next page for the next step!

3. At least one part of the drawing that changes based on the mouse cursor's position

A. The scene changes to night when the mouse is on the bottom half of the screen.



B. Clouds appear when the mouse is on the bottom half of the screen.



C. The UFO flies in the other direction when the mouse is on the bottom of the screen. (Think about what part of the code controls the direction and movement).

D. Anything else you can think of!

When you are done, think about what else you can add to your drawing...Can you make things grow or shrink based on the position? Fade or get brighter? Use your imagination to see what else you can come up with.

Appendix I

PEABODY-INTERVENTION 2: NO CC CONDITIONALS

The following is the curriculum for intervention 2 (testing student preferences concerning creative choice while learning a new concepts, specifically conditionals). This is the set of worksheets (part 1 and part 2) where students could not use creative choice while working with this new concept.

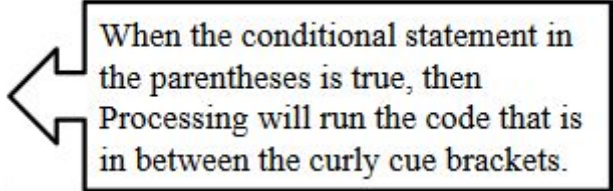
Conditionals Part 1

J. Wilcox, Z. Wood, K. Mork

Introduction: Changing Background Color

A powerful tool we can learn to use in Processing uses a type of statement called a _____. These statements can either be _____ or _____. The way we enter one of these statements into Processing looks like this:

```
if ( _____ ) {
    'This is where we
    Would write the code'
}
```



When the conditional statement in the parentheses is true, then Processing will run the code that is in between the curly cue brackets.

The conditional statement we make will use one of these relational operators:

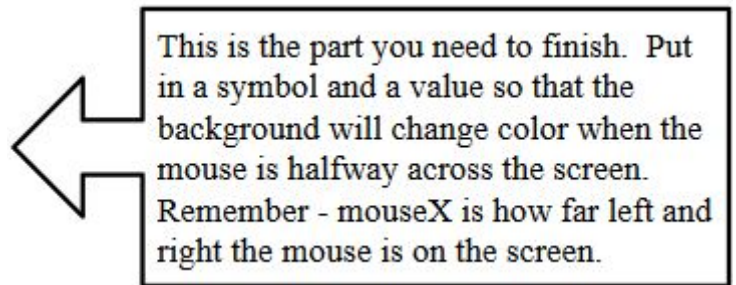
Symbol	Meaning
>	
<	

Finish the following code and type it in.

```
void setup() {
    size(400, 400);
}

void draw() {
    background(50);

    if (mouseX > _____) {
        background(100, 0, 250);
    }
}
```



This is the part you need to finish. Put in a symbol and a value so that the background will change color when the mouse is halfway across the screen. Remember - mouseX is how far left and right the mouse is on the screen.

Now let's try changing more than just background color!

Changing Basic Shapes

Next let's change the shapes that appear on the screen. First, take a look at the following code.

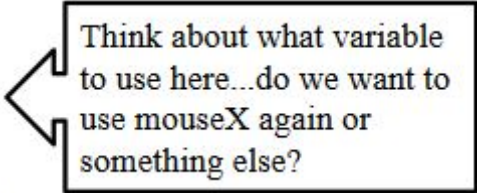
```
void setup() {  
    size(400, 400);  
}  
  
void draw() {  
    background(50);  
    fill(255, 0, 0);  
    ellipse(200, 200, 100, 100);  
}
```

It draws a _____.

See if you can change it to draw a green rectangle when the mouse is on the **top** half of the screen, and draw what it currently draws if the mouse is on the bottom of the screen.

Fill in the following code!

```
void setup() {  
    size(400, 400);  
}  
  
void draw() {  
    background(50);  
    fill(255, 0, 0);  
    ellipse(200, 200, 100, 100);  
  
    if ( _____ ) {  
        fill(0, 255, 0);  
        rect(175, 175, 50, 50);  
    }  
}
```



Think about what variable to use here...do we want to use mouseX again or something else?

What happens now?

Is this what we want to happen? Let's add one more conditional.

```
void setup() {
    size(400, 400);
}

void draw() {
    background(50);

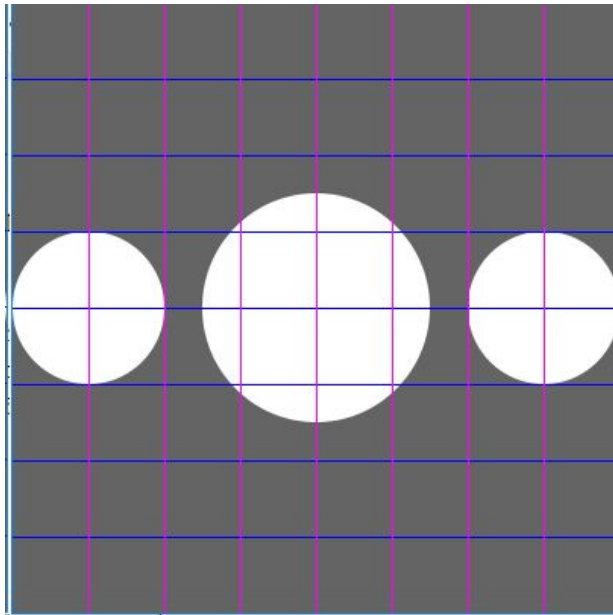
    //draw red ellipse if mouse is on the bottom half of screen
    if (_____ < _____) {
        fill(255, 0, 0);
        ellipse(200, 200, 100, 100);
    }

    //draw a green rectangle if mouse is on the top half of screen
    if (_____ > _____) {
        fill(0, 255, 0);
        rect(175, 175, 50, 50);
    }
}
```

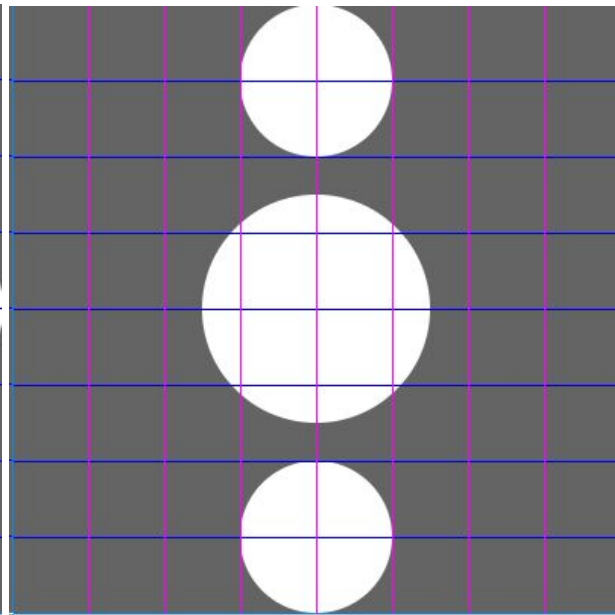
Now let's do a challenge on our own. Switching between a rectangle and ellipse is cool, but what if we did something cooler?

Changing Circle Puzzles

Finish the following code so that when the mouse is on the top half of the screen, the screen will display the first set of ellipses; and when the mouse is on the bottom half of the screen, the screen will display the second set of ellipses. Make your two scenes look like the pictures below:



Ellipses 1



Ellipses 2

Fill in the code on the next page to make “Ellipses 1” when the mouse is on the bottom half and “Ellipses 2” when the mouse is on the top half.

```

void setup() {
    size(400, 400);
}

void draw() {
    background(100);

    //draw ellipses 1 if mouse is on the bottom half of screen
    if (mouseY __ ____) {
        _____
        _____
        _____
    }
    //draw ellipses 2 if mouse is on the top half of screen
    if (______ __ ____) {
        _____
        _____
        _____
    }
}
}

```

For a final challenge, see if you can add more toggles. We change the drawing if the mouse is on the top or bottom of the screen. Let's also change the background color if the mouse is on the left or right half of the screen.

WITHOUT deleting anything, add the following lines of code inside your draw function. (Note: "inside your draw function" means somewhere in between the curly braces after `void draw()`). Do you want to change the background color BEFORE or AFTER you change the drawing? _____

```

//background color if mouse is on left side of screen
if (mouseX __ ____) {
    background(____, __, __);
}
//background color if mouse is on right side of screen
if (mouseX __ ____) {
    background(____, __, __);
}

```

Conditionals Part 2

J. Wilcox, Z. Wood, K. Mork

Recap

In our last session we learned about using conditionals in our code. Remember: a conditional statement is something that can be _____ or _____.

A conditional statement can claim something about a variable. What is a variable?

If our variable changes values, our conditional statement might switch from true to false or from false to true. We can change our actions based on if our conditional is true or not.

So far, our conditional statements have looked like this:

variable < some number

variable > some number

The variables we have looked at are mouseX and mouseY.

Let's think about our variable mouseX again. What does mouseX tell us?

Look at the following conditional statement for a canvas of size(800, 400):

```
mouseX < 400;
```

What does this statement claim? (remember that our canvas has a width of 800).

Is the conditional statement true if the mouse is on the left side of the screen? _____

Is the conditional statement true if the mouse is on the right side of the screen? _____

What conditional statement would claim the mouse is on the right side of the screen?

Variable < or > some number

Remember that we can use conditional statements about mouseX to change our picture. What does the following code do?

```
float px = 0;

void setup() {
  size(800, 400);
}

void draw() {
  background(0, 255, 255);

  if (mouseX < 400) {
    fill(255, 255, 0);
    ellipse(px, 200, 50, 50);
  }
  px = px + 1;
}
```

Statements with Other Variables

We can do a lot by making statements about mouseX and mouseY, but what else can we do? Can we also make statements about other variables? Let's think about what other variables we have used!

Look at the code above and see if you can find any other variables. Another variable our code uses, besides mouseX, is _____. This variable is being used to represent _____

Let's make a statement that the x-position of the ellipse is on the right half of a 800 width canvas. How could I write that statement?

_____ _____ _____
Variable < or > some number

Now, instead of checking if our cursor crosses halfway across the screen, let's check if our ellipse does!

Fill in the code to write: if the ellipse is on the left half of the screen then pick one color, and if the ellipse is on the right half of the screen then pick another color.

```
float px = 0;

void setup() {
    size(800, 400);
}

void draw() {

    //ellipse is on left
    if (_____) {
        background(0, 255, 255);
        fill(255, 255, 0);
    }
    //ellipse is on right
    if (_____) {
        background(0, 0, 0);
        fill(255, 255, 255);
    }

    ellipse(px, 100, 50, 50);
    px = px + 1;
}
```

Practice

Now it's time to work on your own! Try the following puzzles to see what you can do with conditional statements using a combination of variables `px`, `mouseX`, and `mouseY`.

Challenge 1

Fill in the code below to do the following:

1. If the ellipse is on the left half of the screen, make the size of the ellipse grow.
2. If the ellipse is on the right half of the screen, make the ellipse 50 wide and 50 tall.

```
float px = 0;

void setup() {
  size(800, 400);
}

void draw() {
  background(0, 255, 255);
  //ellipse is on left
  if (_____) {
    ellipse(px, 200, ___, ___);
  }
  //ellipse is on right
  if (_____) {
    ellipse(px, 200, 50, 50);
  }

  px = px + 1;
}
```

Challenge 2

Add these conditionals to the draw function to do the following:

1. If the mouse cursor is on the top half of the screen, make the ellipse's color change based on the variable px.
2. If the mouse is on the bottom half of the screen, make it blue

```
//mouse is on top half of screen
if (_____){
    fill(px, 0, 0);
}
//mouse is on bottom half of screen
if (_____){
    fill(0, 255, 0);
}
```

NOTE: This code is setting the ellipse's color - should this code go BEFORE or AFTER the ellipses are drawn?

Bonus Challenges

1. Can you figure out how to make the ellipse's position reset to the beginning of the canvas after it disappears off of the end? This will make the ellipse fly across your screen over and over until you stop the program.
2. Can you change the code you wrote so the ellipse will still grow while it is on the left half of the screen but will also now shrink while it is on the right half of the screen?

Appendix J

PEABODY-INTERVENTION 2: SURVEYS

The following are the variable efficacy and MSLQ survey, variable performance survey, conditional efficacy survey, and conditional performance survey. Together, these make up the four surveys given to the fifth grade classes for intervention 2.

Variables Survey

Answer the following questions honestly - your answers will stay anonymous!

* Required

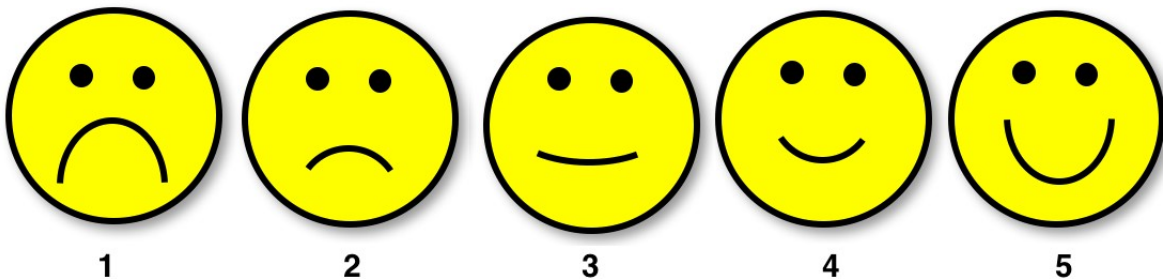
1. Enter Your lunch ID: *

2. Select the Lab Period *

Mark only one oval.

- ☐ Monday 12:45
☐ Monday 1:55
☐ Thursday 8:30
☐ Thursday 9:50

Use this scale to say how confident you are that you are able to do the following now that you've completed the lessons:



3. I can write code to control the speed and direction of an ellipse using variables

Mark only one oval.

1 2 3 4 5

☐ ☐ ☐ ☐ ☐

4. I can figure out how to use variables to change the size of an ellipse

Mark only one oval.

1 2 3 4 5

☐ ☐ ☐ ☐ ☐

5. I can figure out how to use variables to change the position of an ellipse

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. I can figure out how to use variables to change the color of an ellipse

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. I understand variables and can help other students if they are confused about them

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. I can ask for help about variables if I don't understand something in the lessons

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Think about the lessons and worksheets you worked on with variables. What did you make? How hard was it? How creative was it? For the following questions, rate how true you think the following were

9. The worksheets let us be creative

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. The worksheets had us make cool looking pictures/animations

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

11. The lessons gave us a fun challenge*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

12. The instructions were clear and easy to follow*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

13. I was able to finish the assignments*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

14. I had freedom on these worksheets*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

15. I understood variables after completing the worksheets and exercises*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

16. Pick the MOST true of the following:*Mark only one oval.*

- ☐ I wish we had more rigid instructions
- ☐ The worksheets had a good amount of creativity and instructions
- ☐ I wish we had more creativity or freedom in these worksheets

17. Why?

Powered by
 Google Forms

Variable Checkpoint

Answer the following questions to the best of your ability - your answers will stay anonymous!

* Required

1. Enter Your lunch ID: *

2. Select the Lab Period *

Mark only one oval.

☐ Monday 12:45

☐ Monday 1:55

☐ Thursday 8:30

☐ Thursday 9:50

3. A variable is

4. What is the name of the variable that describes how far left or right the mouse cursor is?

5. I can make a variable and name it anything I want (like 'fluffy' or 'px')

Mark only one oval.

☐ True

☐ False

6. Read the following code and select where you think the ellipse will start in your canvas:

```
float px = 0;

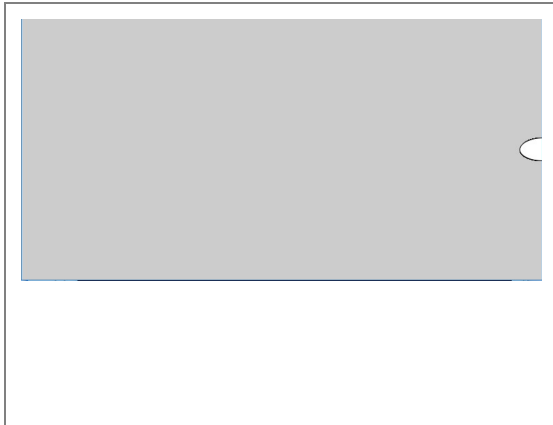
void setup() {
  size(800, 400);
}

void draw() {
  ellipse(px, 200, 70, 35);
  px = px + 1;
}
```

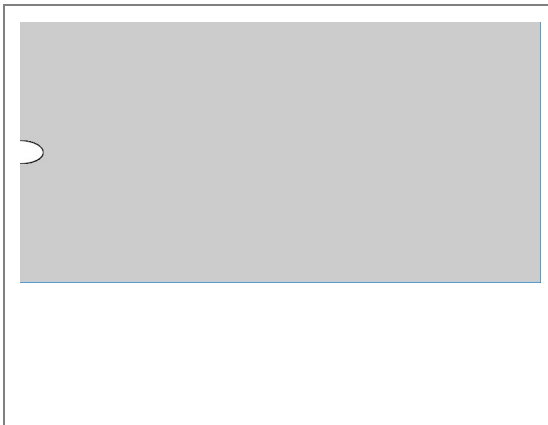
Mark only one oval.



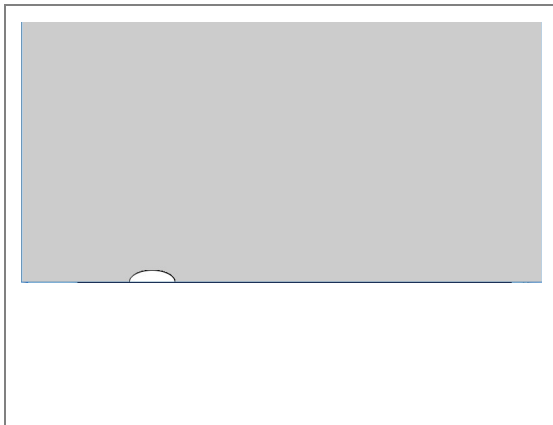
☐ Top of the canvas, partly over



☐ Right side of the canvas, halfway down



☐ Left side of the canvas, halfway down



☐ Bottom of the canvas, partly over

Use this code to answer the next two questions. (notice how the code didn't use very helpful variable names...we should try to give meaningful names to our variables)

```
float fluffy = 0;
float spiderman = 0;
float potato = 0;

void setup() {
  size(800, 400);
}

void draw() {
  fill(spiderman, 0, 0);
  ellipse(fluffy, 400, 70, 35);
}
```

7. To make the ellipse more red with time, I want to add code to the draw function. Which line should I add?

Mark only one oval.

- ☐ fluffy = fluffy + 1;
- ☐ spiderman = spiderman + 1;
- ☐ potato = potato + 1;
- ☐ right = right + 1;

8. To make the ellipse move right, I want to add code to the draw function. Which line should I add?

Mark only one oval.

- ☐ fluffy = fluffy + 1;
- ☐ spiderman = spiderman + 1;
- ☐ potato = potato + 1;
- ☐ color = red;

Conditionals Survey

Answer the following questions honestly - your answers will stay anonymous!

* Required

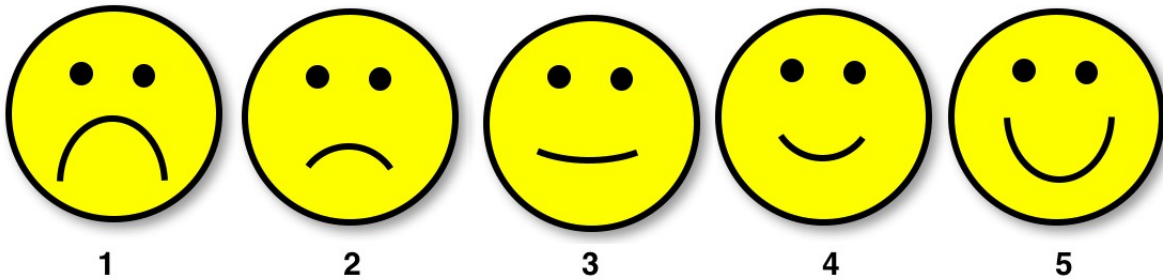
1. Enter Your lunch ID: *

2. Select the Lab Period *

Mark only one oval.

- ☐ Monday 12:45
☐ Monday 1:55
☐ Thursday 8:30
☐ Thursday 9:50

Use this scale to say how certain you are that you are able to do the following. These questions are not about this lab, but just about your experience with fifth grade in general.



3. I can finish my homework assignments by deadlines

Mark only one oval.

1 2 3 4 5

☐ ☐ ☐ ☐ ☐

4. I can get myself to study when there are other interesting things to do

Mark only one oval.

1 2 3 4 5

☐ ☐ ☐ ☐ ☐

5. I can always concentrate on school subjects during class*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. I can take good notes during class instruction*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. I can plan my schoolwork for the day*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. I can organize my schoolwork*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9. I can remember information presented in class and textbooks*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

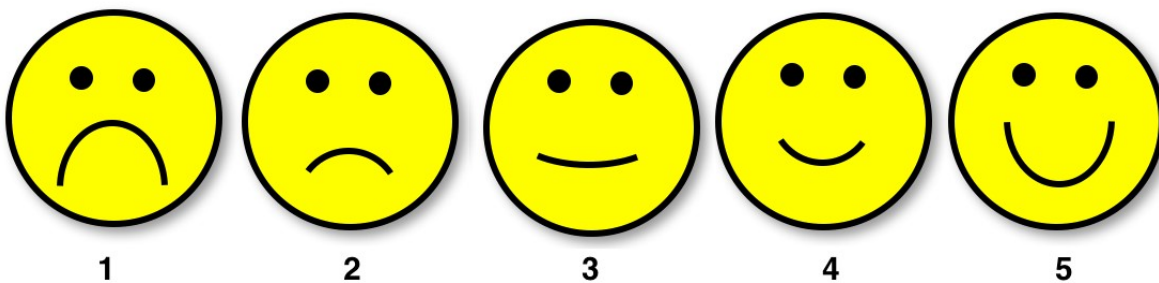
10. I can arrange a place to study without distractions*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

11. I can get myself to do schoolwork*Mark only one oval.*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

The next questions are about computer lab. Use this scale to say how certain you are able to do the following things with conditional statements.



12. I can write conditional statements to check if the mouse is on the left half of the screen

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

13. I can change the picture that shows up on the screen based on a conditional statement being true or false

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

14. I can figure out how to write a conditional statement to check if a moving ellipse is on the left half of the screen

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

15. I can understand conditional statements enough to do the exercises we are given

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

16. I understand the difficult parts of conditional statements beyond just being able to do the worksheets

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

17. I understand conditional statements enough to help other students if they are confused about them

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

18. I can ask for help about conditional statements if I don't understand something in the lessons

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Think about the lessons and worksheets you worked on with conditionals. What did you make? How hard was it? How creative was it? For the following questions, rate how true you think the following were

19. The worksheets let us be creative

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

20. The worksheets had us make cool looking pictures/animations

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

21. **The lessons gave us a fun challenge**

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

22. **The instructions were clear and easy to follow**

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

23. **I was able to finish the assignments**

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

24. **I had freedom on these worksheets**

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

25. **I understood variables after completing the worksheets and exercises**

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

26. **Pick the MOST true of the following:**

Mark only one oval.

- ☐ I wish we had more rigid instructions
- ☐ The worksheets had a good amount of creativity and instructions
- ☐ I wish we had more creativity or freedom in these worksheets

27. **Why?**

Powered by
 Google Forms

Conditional Checkpoint

Answer the following questions to the best of your ability - your answers will stay anonymous!

* Required

1. Enter Your lunch ID: *

2. Select the Lab Period *

Mark only one oval.

- ☐ Monday 12:45
- ☐ Monday 1:55
- ☐ Thursday 8:30
- ☐ Thursday 9:50

3. A conditional statement is *

4. A conditional statement might be true for a while and later the same statement might be false if a condition changed *

Mark only one oval.

- ☐ true
- ☐ false

5. If the screen is size(400, 400), the conditional statement (mouseX < 200) checks if *

Mark only one oval.

- ☐ the mouse cursor is on the top half of the screen
- ☐ the mouse cursor is on the bottom half of the screen
- ☐ the mouse cursor is on the left half of the screen
- ☐ the mouse cursor is on the right half of the screen

6. If the screen is size(400, 400), the conditional statement (mouseY > 200) checks if *

Mark only one oval.

- ☐ the mouse cursor is on the top half of the screen
- ☐ the mouse cursor is on the bottom half of the screen
- ☐ the mouse cursor is on the left half of the screen
- ☐ the mouse cursor is on the right half of the screen

Use this code for the next question

```
int px = 0;

void setup() {
  size(400, 400);
}

void draw() {
  fill(255, 0, 0); //pick red color

  if (mouseX > 200) {
    fill(0, 0, 255); //pick blue color
  }
  ellipse(px, 200, 50, 50);
  px = px + 1;
}
```

7. In the code above, the ellipse is blue if

Mark only one oval.

- ☐ the ellipse is on the left half of the screen
- ☐ the ellipse is on the right half of the screen
- ☐ the mouse is on the left half of the screen
- ☐ the mouse is on the right half of the screen

Use this code for the next question

```
int px = 0;

void setup() {
  size(400, 400);
}

void draw() {
  fill(255, 0, 0); //pick red color

  if (px > 200) {
    fill(0, 0, 255); //pick blue color
  }
  ellipse(px, 200, 50, 50);
  px = px + 1;
}
```

8. In the code above, the ellipse is blue if

Mark only one oval.

- ☐ the ellipse is on the left half of the screen
 - ☐ the ellipse is on the right half of the screen
 - ☐ the mouse is on the left half of the screen
 - ☐ the mouse is on the right half of the screen
-

Powered by



Appendix K

JUVENILE HALL SYLLABUS

This is the syllabus shared with Restorative Partners and the CVA teacher.

CS1: Introduction to 2D Graphics

Kirsten Mork

Overview:

This course is an introduction to computer science/programming through the lens of 2D game design. The goal is twofold: 1) to get students comfortable and excited about the possibilities of computer science, and 2) to teach reusable coding concepts to set students up for success if they choose to take computer science courses in the future. We will work towards these goals by learning how to design and code a simple game. Hopefully by the end each student will have the beginning of a game to show to the class.

(Preferred) Prerequisite skills:

- Sufficient at typing
- Comfortable navigating a computer

Objectives:

- Empower students to believe they can pursue computer science
- Apply logical thinking and creativity to make a game
- Learn computer programming basics/terminology

By the end of the course, students should have a basic understanding of:

- Processing commands
- Functions
- Variables
- I/O
- Loops

Programming Language:

- Processing

Appendix L

CVA TEACHER'S REPORT ON JUVENILE HALL CLASS

Below is the CVA teacher, Greg Murphy's, report on how the class went.

4-22-19

To: Kristen Mork, Eric Mork, & Timothy Wong
Cal Poly SLO Students

From: Greg Murphy
Teacher SLO Co Office of Education
Gmurphy@slocoe.org

I am writing this letter to thank Kirsten Mork, Erick Mork, and Timothy Wong for conducting a computer programming class in the SLO County Juvenile Hall. Over the past two months these Cal Poly students effectively delivered a class on computer programming for 5-8 youth who were housed at the SLO County Juvenile Hall. The class included lessons on coding and taught students how to create a simple video game. All students completed the project at varying degrees of complexity.

Kristen Mork was the primary instructor and designer of the scope and sequence of the class. Eric Mork and Timothy Wong acted as assistants and helped with both lesson delivery and one on one teacher student support. Kristen Mork not only presented well organized and thought out material, but was also able to maintain control of the class and engage students in interesting ways while presenting complex material.

It can be difficult to educate students who live in a juvenile detention facility. Many of them aren't here for being highly engaged learners. I will admit feeling some scepticism when this activity was initially discussed. Regardless of my concerns, this class turned out to be a valuable activity for all participants. The high school students learned many new things, completed a project, were engaged and respectful, and interacted with college students who were working on achieving their own academic goals. I believe that the high school students were successfully exposed to the field of technology and computer programming in a way that was enjoyable and encouraged them to consider exploring this field in the future. The Cal Poly students successfully created and delivered a class in basic computer coding and game design for some of our county's most disadvantaged youth. The activities were successful and "hands on" for both the high school students and the college students.

It is important for people in a community to work to understand the perspectives of individuals they may not normally have access to. "At risk" high school students often do not know many Cal Poly seniors. Cal Poly seniors may not understand the needs and personalities of our community's most disadvantaged teens. By bringing these two groups together we are able to teach both groups not only concrete skills, like computer programming or lesson delivery, but we also teach all involved something about what it is like to be a different person, living in very different shoes. This computer programming class broadened the perspective of all participants and, for this reason, I hope that we can continue conducting classes that involve partnerships between Cal Poly students and SLOCOE students attending classes at the Juvenile Hall.

Thank you,



Greg Murphy MA
Teacher SLOCOE

Appendix M

CVA PRE-COURSE SURVEY

Below is the survey we gave the CVA students at the beginning of our CS course.

Pre Survey

We'd like to ask you a few questions to try and help make this class better. The survey is completely anonymous and won't affect any scores in the class. You don't have to answer any questions you don't want to answer.

1. Number

Select how confident you are of the following (with 1 being not at all confident and 7 being most confident)

2. I can learn art

Mark only one oval.

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

3. I can learn math

Mark only one oval.

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

4. I can learn to use computers

Mark only one oval.

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

5. I can learn computer programming

Mark only one oval.

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

6. I can learn to create art on computers*Mark only one oval.*

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

7. I can learn to create games on computers*Mark only one oval.*

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

8. I can learn to be creative*Mark only one oval.*

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

9. I believe I will receive an excellent grade in this class*Mark only one oval.*

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

10. I'm certain I can understand the most difficult material presented in the worksheets for this course*Mark only one oval.*

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

11. I'm confident I can understand the basic concepts taught in this course*Mark only one oval.*

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

12. I'm confident I can understand the most complex material presented by the instructor in this course

Mark only one oval.

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

13. I'm confident I can do an excellent job on the assignments and projects in this course

Mark only one oval.

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

14. I expect to do well in this class

Mark only one oval.

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

15. I'm certain I can master the skills being taught in this class

Mark only one oval.

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

16. Considering the difficulty of this course, the teacher, and my skills, I think I will do well in this class

Mark only one oval.

	1	2	3	4	5	6	7	
Not at all true of me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very true of me

Answer how strongly you agree or disagree with the following statements:

17. I would consider a career in computer science

Mark only one oval.

	1	2	3	4	5	6	7	
disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

18. I will be able to get a job in computer science one day if I want to

Mark only one oval.

	1	2	3	4	5	6	7	
disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

19. I would like to learn computer programming

Mark only one oval.

	1	2	3	4	5	6	7	
disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

20. I would like to learn how to make games on computers

Mark only one oval.

	1	2	3	4	5	6	7	
disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

21. I am a creative person

Mark only one oval.

	1	2	3	4	5	6	7	
disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

22. I like it when my classes let me be creative

Mark only one oval.

	1	2	3	4	5	6	7	
disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

Powered by

