

Winter 1-10-2017

Gamification Through Algebraic Coding

Rachel L. Serpe
rlserpe1@gmail.com

Follow this and additional works at: http://digitalcommons.brockport.edu/ehd_theses

To learn more about our programs visit: <http://www.brockport.edu/ehd/>

Repository Citation

Serpe, Rachel L., "Gamification Through Algebraic Coding" (2017). *Education and Human Development Master's Theses*. 711.
http://digitalcommons.brockport.edu/ehd_theses/711

This Thesis is brought to you for free and open access by the Education and Human Development at Digital Commons @Brockport. It has been accepted for inclusion in Education and Human Development Master's Theses by an authorized administrator of Digital Commons @Brockport. For more information, please contact kmyers@brockport.edu.

Gamification Through Algebraic Coding

by
Rachel Leigh Serpe

A thesis submitted to the College at Brockport, State University of New York, in partial fulfillment of the requirements for the Degree of Master of Science in Education.

January 10th, 2017

Table of Contents

ABSTRACT.....	2
INTRODUCTION.....	2
A. PURPOSE.....	3
LITERATURE REVIEW	3
A. GAMIFICATION	4
B. BOOTSTRAP CURRICULUM	8
RESEARCH QUESTION	13
METHOD.....	13
A. SAMPLE.....	14
B. VIE THEORY	14
C. IMPLEMENTATION OF MOTIVATIONAL SURVEY.....	17
RESULTS.....	18
CONCLUSION.....	21
REFERENCES	22
APPENDICES.....	24
A. Appendix A – Motivational Survey and Construct Analysis.....	24
B. Appendix B – Algebra I Common Core Standards used in this Unit.....	26
C. Appendix C – Example of a content lesson and its following application lesson.....	27

Abstract

From an urban high school in upstate New York, gamification was introduced through coding to teach an Algebra I unit. The Value Instrumentality Expectancy (VIE) Theory was used to measure motivation to determine if learning coding by gamifying a unit and applying it in the computer lab motivated students to learn Algebra I content. There was a significant increase in each motivational construct. This implies that if teachers dedicate themselves to learn coding and the pedagogical knowledge needed to teach a gamified unit, then there can be an increase in motivation to learn Algebra I content.

Introduction

Motivation is an issue in mathematics classrooms today. When high school students have low motivation, they can fall behind and miss underlying concepts leading to poor achievement in math (Beswick, 2010). Beswick (2010) states that students who struggle with mathematics often lose interest at a young age, and therefore teachers need to find ways to engage their students and increase their motivation. In today's digital generation, gamification has become a popular tactic to encourage specific behaviors and increase motivation (Hsin-Yuan Huang & Soman, 2013). Gamification is defined as "the use of game designed elements and game mechanics in non-game contexts" (Dominguez et al., 2012, p. 380). Faghihi et al. (2014) present the idea that video game element techniques can be used to teach mathematical concepts in the classroom. These concepts should be explained in a way that helps learners "make a connection between the mathematical concepts and their real life experience" (Faghihi et al., 2014, p. 182).

Purpose

The purpose of this project is to introduce gamification to high school Algebra I students through algebraic coding to determine the impact on student motivation. There is research on gamification, and research on mathematics motivation, but there is not research on mathematics motivation through gamification by algebraic coding. Schanzer et al. (2015) describes an approach to a teaching design that measurably improves student performance on algebraic word problems. Their approach is a curriculum called Bootstrap which is used in both middle and high-school math and computing classes across the USA. Bootstrap simultaneously teaches students how to design programs and solve algebraic word problems through video game programming. This project adds to the body of knowledge of gamification by informing others of motivation changes from high school students who learned Algebra I content to apply their knowledge and develop an interactive game through coding.

Literature Review

Motivation and engagement are prerequisites for the completion of a task or encouragement of a specific behavior (Ritchel, 2010). The reasons for low performance in math include boredom, lack of engagement, absenteeism, and being distracted by technology (Ritchel, 2010). Current research on math education gives evidence for new strategies which motivates math students and encourages them to problem solve and think critically.

Gamification

In today's digital generation, gamification has become a popular tactic to encourage specific behaviors and increase motivation and engagement (Hsin-Yuan Huang & Soman, 2013). Though commonly found in marketing strategies, gamification is now being implemented in many educational programs, helping educators find the balance between achieving their objectives and catering to evolving student needs.

In their work, Faghihi et al. (2014) present the idea that video game element techniques can be used to teach mathematical concepts in the classroom. These concepts should be explained in a way that helps learners "make a connection between the mathematical concepts and their real life experience" (Faghihi et al., 2014, p. 182). Gamification is defined as "the use of game designed elements and game mechanics in non-game contexts" (Dominguez et al., 2012, p. 380). Dominguez et al. (2012) believe that gamification can be used as an educational tool to increase student motivation. To have a better understanding of the research presented, one must first know the background of gamification. Video games and the gaming industry have increased in popularity drastically over the last 20 years (Dominguez et al., 2012). New games and apps are created daily with the intent to expand the gaming industry and hook new users. Facebook has an entire social gaming system just for its users, and systems such as the Wii and Kinect aim to incorporate more of a family aspect to gaming. With these games and the rise of new technology, it was only a matter of time before researchers and teachers thought to connect gaming and education. Educational researchers, per Dominguez et al., have been especially interested in gaming entertainment (2012). "Video games are interactive activities that continually provide challenges and goals to the players, thus involving them into an active

learning process to master the game mechanics” (Costner, 2005, p. 110). Due to the high involvement and addictive nature that video games create, research has been done to try and discover what makes video games so popular and how this can be used successfully to motivate students in school to become more interested in topics being taught (Dominguez et al., 2012). Most researchers agree it is beneficial to use certain aspects of video games and relate them in a non-gaming educative context (Dominguez et al., 2012). This concept is more frequently known as gamification, and its main objective is to increase user experience and engagement (Dominguez et al., 2012). Like videogames, gamification is based on technology. Therefore, researchers define gamification as “incorporating game elements into a non-gaming software application to increase user experience and engagement” (Dominguez et al., 2012, p.381).

To create an increase in student motivation through a gamified system, it is essential to focus on the fundamental elements that make video games so appealing to their players (Dominguez et al., 2012). Why do millions of users continue to play games? What is so addicting about a game? Lee and Hammer (2011) suggest that games are motivating because they impact users on a social, emotional, and cognitive level. Therefore, gamification should also focus on these three areas to hook students in the classroom. In the cognitive area, “a game provides a complex system of rules along with a series of tasks that guide players through a process to master those rules” (Dominguez et al., 2012, p. 381). These tasks allow players to repeatedly try and fail, and users cannot succeed until the necessary skills are acquired. Game design often encourages players to experiment without fear of causing irreversible damage by giving them multiple lives, or allowing them to start again at the most recent checkpoint. Incorporating this “freedom to fail” into classroom design is noted to be an effective dynamic in

increasing student engagement. If students are encouraged to take risks and experiment, the focus is taken away from results and re-centered on the process of learning instead (Stott & Neustaedter, 2013). The effectiveness of this change in is recognized in modern pedagogy by the increased use of formative assessment in today's classroom. Like the game dynamic of having the 'freedom to fail', formative assessment focuses on the process of learning rather than the result by using assessment to inform subsequent lessons and separating assessment from grades whenever possible. As Kapp (2012) notes, this doesn't mean letting students have four chances at a multiple-choice question with four possible answers. What it means is "encouraging learners to explore content, take chances with their decision making, and be exposed to realistic consequences for making a wrong or poor decision" (p. 65).

Freedom to fail stimulates an emotional area of the player, having them deal with both success and failure. On one hand, players can complete tasks and have positive emotions. They can then be awarded with points, trophies, unlocked items, etc. which makes players feel accomplished. On the other hand, when players fail they often have feelings of anxiety and frustration. Csikszentmihalyi (2008) claims that if the difficulty of tasks is correctly balanced, then it can drive the players "to a flow state that his highly motivating" (p.382). Therefore, in the classroom, it is imperative for educators to assign tasks that are appropriate for the given population. If multiple players interact through the game, these interactions then impact a player's social area. Players can work together to complete a certain task or they can be competing against a specific user. These interactions allow players to take on a game identity (Dominquez et al., 2012). The idea behind gamification is to apply the cognitive, emotional, and

social ideas used in gaming and use these to design educational lessons to make learning more motivating for the student. (Dominquez et al., 2012).

Per Muntean (2011), the main purpose of gamification is to combine intrinsic motivation with an extrinsic one to raise motivation and engagement. Intrinsic motivations come from within the user causing them to make a decision (Muntean, 2011). Extrinsic motivations occur when something or someone else determines the user to make an action (Muntean, 2011). Intrinsic motivations would be competition, cooperation, sense of belonging, love or aggression, while extrinsic motivations would be levels, points, badges, awards, missions (Viola, 2011). By using gamification in e-learning, Muntean (2011) believes one can trigger a more efficient and engaging learning behavior from a student. B.J. Fogg (2002) argues that people sometimes respond to computers as if they were a person, especially when gaming. Fogg (2002) claims that to trigger a desired behavior within students, they need to be motivated and should be given tasks that are appropriate for them, tasks that they can solve. Stott and Neustaedter (2013) note that "the tricky part, and the part that is ultimately at the core of the experience, is identifying intrinsic rewards relative to the culture of the local community that one is seeking to engage, and building game-like interactions on top of those."

JB Fogg (2002) studied the concept of persuasive technology and how people can design systems that impact the user on an affective level. He proposed the Fogg Behavior Model (FBM) that studies the factors that go into generating a certain behavior. The model is comprised of 3 main elements: motivation, ability and triggers. Motivation is needed for a student to complete a desired task. A student may be able to solve a problem, but if he has little or no motivation to do so, he probably won't complete the task. Fogg (2002) states that once a student's social

reputation is at stake or he is conscious of the fact that he might get a low grade, the motivation, either positive or negative will determine him to solve the problem. Ability is also a factor that influences the occurrence of a behavior. A student can be highly motivated and willing to work, but a certain behavior cannot occur if he or she does not have the ability to solve the task. Muntean (2011) also claims that sometimes high motivation can allow a student to find the means to accomplish a task, and in turn gain the ability to solve a task. According to Fogg (2002), motivation and ability alone are not enough to determine a behavior. A certain behavior needs a “trigger”, something to tell the user to complete the action in a certain moment, also referred to as a “call to action” (Fogg, 2002). These triggers are directly connected to motivation, and can be a spark, a facilitator, or a signal (Fogg, 2002). A spark is something that tends to motivate a user. A facilitator can offer ability to highly motivated users, and lastly a signal is when users have both ability and motivation, but they need a little reminder. Fogg (2002) states that when motivation, ability, and a trigger occur at the same time, a target behavior can now occur.

Bootstrap Curriculum

Many educators have tried to use video game technology to help improve students’ achievement in mathematics. More recently, educators have started to use computer programming to teach mathematics. Schanzer, Fislser, Krishnamurthi and Felleisen (2015) believe that performance gains in programming have come up short. Schanzer et al. (2015) claim that these efforts fail to align the computing and math concepts at a level required to truly achieve transfer of learning. Schanzer et al. (2015) state that “transfer of learning between

domains typically requires both deep structural connections between the domains and explicit instruction in how to apply concepts from one discipline in the other” (p. 1). Many computer projects fail to handle one of these two requirements when trying to transfer mathematical problem-solving to students. The research of Schanzer et al. (2015) describes an approach to a teaching design that measurably improves student performance on algebra word problems. Their approach is a curriculum called Bootstrap which is used in both middle and high-school math and computing classes across the USA. Bootstrap simultaneously teaches students how to design programs and solve algebraic word problems through video game programming.

Bootstrap uses the addictive nature of videogames and students' interest around videogames to teach algebraic concepts through programming. Bootstrap takes students' excitement and confidence around gaming and directly applies it to algebra to have students create a videogame independently. In a video game, the players themselves make things happen. Players don't just consume what the game designer has placed before them but instead decide how to create their game, pick their choices and see that their decisions now matter and affect the outcome of their game. Creating a video game using algebraic coding is considered an authentic task. Per Beswick (2010), ‘authentic’ and ‘situated’ math problems are used to convey something stronger than just talking about ‘real-life’. Beswick (2010) argues that the goal of authentic problems is to develop “flexible mathematical concepts able to be used in whatever context when and as required” (p. 377). ‘Authentic’ problems allow students to be creative, come up with alternate solutions, and there may not be one right answer. There is no “ready-made algorithm” or a specific formula students just plug into. ‘Authentic’ tasks involve constant problem solving, allow students to be innovators and challenge their own thinking.

'Authentic' tasks should create students who become problem solvers, who can then use this skill outside of school to be successful in the real world. At the end of their algebra unit, students will have a completed workbook filled with word problems, notes and math challenges, as well as a videogame of their own design, which they present and then can share with friends and family. Bootstrap applies mathematical concepts and rigorous programming principles to create a simple videogame, while being aligned with the Common Core Standards. A student videogame consists of 3 characters involving a player, a target and a danger. They design what each character looks like, pick their background, and then use algebraic concepts to see if or when their characters will collide, figure out how characters move and interact, and learn to update and constantly fix their coding. Some math topics learned (or re-visited) in this unit include: the coordinate plane, midpoint and distance formula, order of operations, variables, functions, input/output, domain and range, function composition, inequalities, piecewise functions, Pythagorean Theorem, and number lines. According to Schanzer et al. (2015), students can start to understand these concepts more because they don't just study vocab, they learn to enter an input into their code to get anything to change in their output.

The Bootstrap project is carefully designed so that each game feature teaches a particular mathematical concept. "Each unit in the curriculum has three integrated components: a new feature for students to add to their games, a new programming construct or concept needed to implement the feature, and an underlying mathematics concept that relates to the programming concept" (Schanzer et al., 2015, p.3). Educators should use Bloom's Taxonomy as a guide in designing the progression of the course. Students should complete a level or assignment to the educators liking before being able to progress to the next. As an

educator, one can do this incorporating lower order thinking skills into the first stages (identifying, remembering, understanding), progressing to higher order thinking skills in subsequent levels (analyzing, evaluating, critiquing, summarizing) and finally arriving at the highest order thinking skills in the final levels (composing, creating, designing, planning, inventing) (Bloom, 1956). This is an intelligent design because the students will need the knowledge they gained in earlier stages to successfully complete the higher levels of their game. In Bootstrap, students first start by creating a player, a target and a danger. Students then go on to learn how to program their characters to move left and right. Their danger and target are then detected if they go off screen by checking with student set inequalities that bound the characters to the given coordinate grid. An element has gone off-screen when its x-coordinate falls outside of the screen boundaries. Bootstrap uses this function as an opportunity to practice function composition, by having students write separate functions to check each of the left and right edges of the screen. The platform also consults student-supplied functions to see if the player collided with the target or the danger by using the distance formula. If game elements have collided, the score is either raised (if the player collected the target) or the game is ended (if the player collides with the danger). Bootstrap allows students to learn and explore inequalities, Pythagorean theorem, distance formula, order of operations, functions, domain, and range, all by seeing how their inputs can impact their video game. Students learn to troubleshoot errors, fix mistakes, and update their code to get a wanted result. Kapp (2012) states that "feedback is a critical element in learning. The more frequent and targeted the feedback, the more effective the learning" (p. 65). Feedback is already a key element in education even without any attempts to integrate game design, but

Kapp (2012) notes that educators can increase feedback mechanisms by harnessing elements of game design through "continual feedback to learners in the form of self-paced exercises, visual cues, frequent question-and-answer activities, a progress bar, or carefully placed comments by non-player characters" (p. 65). In a traditional algebra classroom, students can hand in classwork, homework, or a quiz and may not get feedback for days depending on when the teacher goes through and grades them. Sometimes, the teacher may not provide feedback and the student may have incorrect answers but get credit for doing the work. Through Bootstrap, students are continually given feedback on their code. Their game will not run unless their code is correct. Students learn to independently troubleshoot their own problems, and learn by trial and error. The freedom to fail concept in games has direct links to the concept of formative assessment in pedagogy; both incorporate ongoing assessment and feedback that is separated from permanent grades. "Rapid feedback in games has direct links to formative assessment in the same way. The concept of designed progression in games has direct links to the concept of scaffolded learning in pedagogy; both structure learning in carefully planned increments in order to increase engagement" (Stott and Neustaedter, 2013, p. 7).

In a traditional learning environment, a student's motivation to learn effectively can be hindered due to several reasons. However, with the successful application of gamification techniques, the delivery of the information can transform a simple or mundane task into an addictive learning process for the students. Through gamification, teachers can direct their classroom environment towards raising both engagement and achievement. As with any pedagogical framework, the literature suggests that an educator must be careful to consider the context in which they are teaching: who their students are, and what the shared goals of

the class are. While the underlying objective of applying gamification to any education program is to prompt some type of behavioral change in the student, many instructors specifically look to tackle the issue of student motivation and engagement during their learning process. For students, gamification serves the purpose of minimizing negative emotions that they usually encounter in traditional forms of education. Gamification lets students get the knowledge and skills needed, using the freedom to fail technique in gaming environments, without the embarrassment factor that usually forms a part of classroom education (Hsin-Yuan Huang & Soman, 2013). Instructors can efficiently achieve their objectives and use tracking mechanisms through gaming to get feedback on their students' progress.

Research Question

Will gamification through algebraic coding have a positive effect on students' motivation to learn Algebra I mathematics?

Method

This was actual classroom research, implying of the many dynamics that the teacher and researcher had to consider. Additionally, this research is a final project for a master's thesis, indicating the limitations determined by an Institutional Review Board (IRB). In the midst of the messiness of classroom events and research (Gravemeijer & Cobb, 2006), a design experiment was used to answer the research question. Motivation for this work was guided from the

Valence Instrumentality Expectancy (VIE) Theory. Motivation was measured by administering a pre-survey before any gamification through algebraic coding (through Bootstrap curriculum) takes place. After the intervention (Algebraic Coding unit in Algebra class using Bootstrap curriculum), the students were administered a post-survey and data was evaluated to see if gamification through algebraic coding increased motivation of the Algebra I content in this unit.

Sample

From a high school in upstate New York that offers seven sections of Algebra I classes, one class was chosen to participate in this research. In a sense, this was a convenience sample based on the enrollment in the teacher and researcher's one Algebra I class. There were 15 students between ages 15 and 18 years old that participated in this study. Each student was given a random personal identifier to match the pre-post motivation surveys for analysis. The diversity of the class included seven males and eight females. Of these 15 students, six had Individualized Education Plans (IEPs). In addition, a different group of six students had Academic Intervention Support (AIS) for learning English. The breakdown of race consists of five African American students, four Hispanic students, and six Caucasians.

VIE Theory

The Latin root of the word "motivation" means, "to move" which is appropriate for this study since their video game actively sought to move students toward increased interest in

mathematics and coding. Teachers who aspire to engage students in opportunities for deep understanding of curricular content must be concerned with motivational issues which address how students engage in and persist at challenging but meaningful content (Blumenfeld et. al., 1991). The expectancy theory of motivation fits nicely in the classroom with real-life learning experiences that require students to think through challenging problems.

Expectancy theory is “highly cognitive in orientation and implies that individuals calculate the desirability of various outcomes and act accordingly” (Landy & Becker, 1985). Motivation theories in this genre include valence theory, instrumentality theory, valence-instrumentality-expectancy (VIE) theory, utility theory, and value-expectancy theory (Landy et. al., 1985). Each approach assumes that individuals are, for the most part, rational and seek to maximize gains while minimizing losses. This framework is appropriate for this study since it links performance, persistence, and choice directly to expectancy-related (“Can I do this?”) and task-value (“Do I value this?”) beliefs (Eccles & Wigfield, 2002).

The theoretical perspective of the motivational survey used in this study was based on the Value Instrumentality Expectancy (VIE) theory. The survey instrument came from the 2009-2010 *Attitude of High School Students Toward a Scientific or Engineering Major Survey (Attitude STEM Survey)*. The *Attitude STEM Survey* was adapted from a survey created by Switzer and Benson for use with engineering undergraduate students (Switzer & Benson, 2007) and is now in the third iteration. The current survey was based on the original work by Switzer, Benson, and Wade but was created to measure pre-post math and coding motivation of high school Algebra I students.

VIE theory conceptualizes motivation as the interaction of three elements: valence, instrumentality, and expectancy. Valence (V) is the perceived value of the behavior and its goals. Valence may be produced by the value of the behavior (VB) such as it is fun or interesting, or it can originate from the value of the goals (VG) associated with the behavior. In this study, the value of the behavior was participating in high school mathematics and coding. How much did the students value the process of their coursework and the program? The value of the goal was being successful in math or coding. How much did the students value math and coding professions? As valence increases motivation increases. For example, a student may place a high value on earning good grades (goal) for the potential to earn a job in math or computer science (another goal) yet the required schoolwork (the behavior) is not always enjoyable. Another way to increase the valence of hard work is for the teacher to make the content meaningful. If the students understand that what they are learning has meaning in an authentic way, they are more likely to participate in the behaviors conducive to academic success.

Instrumentality (I) is the perceived strength of connection between the behavior and the goals. In this study, instrumentality measured if students felt they were successful in high school, then they would be successful in math or coding in the future. For motivation to exist the student must believe that performing the behavior results in progress towards achieving the desired goals. If the student working hard in mathematics class learns that their state no longer offers math or coding jobs based on high school or college performance, it will be perceived that the hard work has no utility, or instrumentality.

Expectancy (E) is the perceived probability that the behavior can be successfully performed. In this study, expectancy measured if students felt confident they were getting the skills, knowledge, and resources to successfully learn mathematics and coding in high school. Students must believe they have available the needed knowledge, skills, and resources to successfully perform the behavior. The more a student expects the content can be learned the higher the student's motivation will be.

According to VIE Theory, all three elements—valence, instrumentality, and expectancy—must be present for motivation to exist. For a person to want to perform a behavior, the behavior must be associated with one or more goals and there must be value coming from the behavior itself and/or one or more of the goals. There must also be perceived progress toward the goal, and a belief that the individual is capable of the behavior. If motivation is present then all three constructs must have some measurable value, but if one or more of these constructs does not exist, then there is no motivation (Eccles & Wigfield, 2002).

Implementation of the Motivational Survey

A pre- and post-survey of the same 16 questions was given to the 15 students to measure their motivation to learn mathematics before and after the intervention. Instructions for the surveys were printed right on the pre- and post-survey handed out. Students circled the answer that best represented their opinion of the question asked. Students had the opportunity to choose a Likert Scale that they most agreed with. Both pre- and post-surveys were administered during Algebra class time and took at most 10 minutes for each survey. Students were randomly assigned a number (1 up to 15) to place on their survey. Numbers

were given out based on random a seating chart. Students kept the same number on their pre- and post-surveys for data research. The surveys were collected and placed into a confidential folder that only the researcher and research advisor had access to. Data was then analyzed using General Linear Model (GLM) repeated measures. Each survey question was mapped to a specific construct which was determined from adapting the previous Value Instrumentality Expectancy (VIE) constructs from the 2009-2010 Motivational Survey constructs from Clemson University.

Results

In an Algebra I classroom with 15 students who participated in the study, 10 lessons on functions were taught using coding. There were 15 Common Core State Standards (CCSS) covered throughout this Algebra I unit. Students were taught the content of coding, and were then taken to the lab to apply what they learned. Before the students could be successful in their coding, they had to first understand the Algebra I content. The students appeared to find the application of the content motivational because of the cycle of learning and applying it in the computer lab. See Appendix C for an example of both the CCSS covered in one content based lesson and the following application lesson in the computer lab.

For the 16 survey questions, a numerical value was assigned to each of the five answer choices. A five was assigned for strongly agreed (SA), a four was assigned for agreed (A), a three was assigned if neutral (N), a two was assigned if disagreed (D), and a one was assigned if strongly disagreed (SD). Data from the survey answers were then imported into SPSS software for each of the 16 students for both their pre-survey and their post-survey results. The surveys

measured the mean pre- post means for the values of the goal (VG), value of the behavior (VB), instrumentality (I) and expectancy (E). Table 1 shows that the post-means were all higher for the pre-means for all four constructs.

Table 1: Pre-Post Means for the constructs

		Mean	N
Pair 1	preEmean	3.2500	15
	postEmean	3.8167	15
Pair 2	preVGmean	2.9333	15
	postVGmean	3.7667	15
Pair 3	preVBmean	2.9500	15
	postVBmean	3.8667	15
Pair 4	preImean	3.3833	15
	postImean	3.8167	15

Whenever a statistical analysis is performed and results interpreted, there is always a probability that the results are purely by chance. However, the probability that the process was simply a chance encounter can be calculated, and a minimum threshold of statistical significance can be set. If the results are obtained such that the probability that they are simply a chance process is less than this threshold of significance, then we can say the results are not due to chance. In this study, the distribution of parameters follows a normal distribution. The cut-off value for a two tailed t-test for this study was set at a .05 threshold using percentage points of the T distribution. An ANOVA with repeated measures was used to compare the four group pre-post motivational means across time to see if the intervention had a significant effect on motivation. The significance of the values from Table 2 are as follows: value of the goal (VG) was 0.004; value of the behavior (VB) was 0.000; instrumentality (I) was 0.021; and expectancy (E) was 0.011. This shows that all four constructs indicate a significant change in motivation because of the intervention and not by random chance.

Table 2: Tests of Within-Subjects Contrasts (repeated measures ANOVA)

Source	Measure	prepost	Type III Sum of Squares	df	Mean Square	F	Sig.
prepost	VG	Level 1 vs. Level 2	10.417	1	10.417	12.069	.004
	VB	Level 1 vs. Level 2	12.604	1	12.604	21.175	.000
	I	Level 1 vs. Level 2	2.817	1	2.817	6.789	.021
	E	Level 1 vs. Level 2	4.817	1	4.817	8.636	.011

Cronbach's alpha (See Table 3) was computed to determine if the survey used with multiple Likert scale questions was reliable. Since the surveys questions were designed to measure latent variables (that are often hidden or unobservable), they are often difficult to truly measure. Cronbach's alpha informs that the four constructs have high reliability.

Table 3: Cronbach's Alpha Test of Reliability

Reliability Statistics			Cronbach's alpha	Internal consistency
Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items	$\alpha \geq 0.9$	Excellent
.857	.858	8	$0.9 > \alpha \geq 0.8$	Good
			$0.8 > \alpha \geq 0.7$	Acceptable
			$0.7 > \alpha \geq 0.6$	Questionable
			$0.6 > \alpha \geq 0.5$	Poor
			$0.5 > \alpha$	Unacceptable

Conclusion

How to teach so students are motivated to learn Algebra I can be a complex task for teachers. Gamification was one method this teacher researcher used to increase motivation to learn mathematics. In an Algebra I classroom in upstate New York, a pre-post survey was administered before and after an experimental treatment of a gamification unit. This should be considered a pilot study that allows Algebra I teachers to have confidence that using gamification through coding in their class, on a small scale level, motivates students to learn mathematics. However, the teacher researcher of this study has worked for three years to develop her own pedagogical content knowledge using coding as a tool to teach algebra. Thus, teachers content knowledge is believed to be critical for the outcome of students increased motivation. This study was a confirmation that by teachers developing an understanding of coding, that they can teach the standards that their students are responsible to learn in a way that motivates learning.

References

- Beswick, K. (2010). Putting context in context: An examination of the evidence for the benefit of 'contextualized' tasks. *International Journal of Science and Mathematics Education, 9*(2), 367-390.
- Bloom, B. S. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals*. New York: David McKay.
- Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational psychologist, 26*(3-4), 369-398.
- Csikszentmihalyi, M. (2008). *Flow: The psychology of optimal experience*. New York: HarperCollins.
- Dominguez, A., Saenz-de-Navarrete, J., de-Marcos, L., Fernandez-Sanz, L., Pages, C., & Martinez-Herraiz, J. J. (2012). Gamifying learning experiences: Practical implications and outcomes. *Computers & Education, 63*(2013), 380-392.
- Eccles, J. S., & Wigfield, A. (2002). Motivational beliefs, values, and goals. *Annual review of psychology, 53*(1), 109-132.
- Faghihi, U., Brautigam, A., Jorgenson, K., Martin, D., Brown, A., Measures, E., & Maldonado-Bouchard, S. (2014). How Gamification Applies for Educational Purpose Specially with College Algebra. *Procedia Computer Science, 41*, 182–187.
- Fogg, B. J. (2002). Persuasive technology: Using computers to change what we think and do. *Ubiquity 2002*, 89-120.

Gravemeijer, K., & Cobb, P. (2006). Design research from a learning design perspective. *Educational design research*, 17-51.

Hsin-Yuan Huang, W., & Soman, D. (2013). A Practitioner's Guide to Gamification of Education.

Kapp, K. M. (2012). Games, Gamification, and the Quest for Learner Engagement. *T+D*, 66(6), 64-68.

Kostner, R. (2005). *A theory of fun for game design*. Scottsdale, Arizona: Paraglyph Press.

Landy, F. J., & Becker, W. S. (1985). *Adaptive motivation theory* (No. 85-1). PENNSYLVANIA STATE UNIV UNIVERSITY PARK DEPT OF PSYCHOLOGY.

Lee, J. J., & Hammer, J. (2011). Gamification in education: what, how, why bother? Definitions and uses. *Exchange Organizational Behavior Teaching Journal*, 15(2), 1-5.

Muntean, C. (2011). Raising engagement in e-learning through gamification. The 6th International Conference on Virtual Learning ICVL 2011, 324-349.

Richtel, M. (2010). Growing up digital, wired for distraction. *The New York Times*, 1.

Schanzer, E., Fisler, K., Krishnamurthi, S., & Felleisen, M (2015). *Transferring Skills at Solving Word Problems from Computing to Algebra through Bootstrap*. SIGCSE'15, March 4–7, 2015, Kansas City, MO, USA.

Stott, A., & Neustaedter, C. (2013). Analysis of gamification in education. *Surrey, BC, Canada*.

Viola, F. (2011). *Gamification: I Videogiochi nella vita quotidiana*. Italy: Arduino Viola.

Appendix A – Motivational Survey and Construct Analysis

2016-2017 Student Survey

Demographic Section:

Mark the best choice with an X on the correct blank.

1. Gender Male Female Other
2. I want to attend a 2- or 4- year college or university Yes No
 If Yes in number 2 above, do you want to major in:
 Math Computer Science Other

Survey:

Indicate how strongly you agree or disagree with each of the following statements by circling the best representative of your opinion.

SA=Strongly Agree A=Agree N=Neutral D=Disagree SD=Strongly Disagree

- | | |
|--|-------------|
| 1. I am developing problem-solving skills. | SA A N D SD |
| 2. I would be happy if I was successful in math class. | SA A N D SD |
| 3. I am confident when I answer questions in my math class. | SA A N D SD |
| 4. I get satisfaction from doing well in my math class. | SA A N D SD |
| 5. I solve real world problems in math class. | SA A N D SD |
| 6. My parents would be proud if I learned how to code. | SA A N D SD |
| 7. Learning high school math is important to be successful in college. | SA A N D SD |
| 8. I am confident I can be successful in math class. | SA A N D SD |
| 9. I would be proud of myself if I could write code in my math class. | SA A N D SD |
| 10. I like solving math problems that involve real life situations. | SA A N D SD |
| 11. I can learn the skills necessary to be successful in mathematics. | SA A N D SD |
| 12. I enjoy learning in my math class. | SA A N D SD |
| 13. I would get respect from knowing how to code in high school. | SA A N D SD |
| 14. It is important that I understand mathematical concepts. | SA A N D SD |
| 15. I want to keep learning new things. | SA A N D SD |
| 16. I get satisfaction from solving real world problems in math class. | SA A N D SD |

2016-2017 Mapping of Items to Constructs

Value of the Behavior (VB) -Participating in high school Math or Coding

How much do the students value the process of their coursework / program?

- 4. I get satisfaction from doing well in my math class.
- 10. I like solving math problems that involve real life situations.
- 12. I enjoy learning in my math class.
- 16. I get satisfaction from solving real world problems in math class.

Value of the Goal (VG) - Being successful in Math or Coding

How much do the students value Math and Coding professions?

- 2. I would be happy if I was successful in math class.
- 6. My parents would be proud if I learned how to code.
- 9. I would be proud of myself if I could write code in my math class.
- 13. I would get respect from knowing how to code in High School.

Instrumentality (I)

If I am successful in high school, then I will be successful in Math or Coding in the future.

- 3. I am confident when I answer questions in my math class.
- 8. I am confident I can be successful in math class.
- 11. I can learn the skills necessary to be successful in mathematics.
- 15. I want to keep learning new things.

Expectancy (E)

The students feel confident they are getting the skills, knowledge, and resources to successfully learn mathematics and coding in high school.

- 1. I am developing problem-solving skills.
- 5. I solve real world problems in math class.
- 7. Learning high school math is important to be successful in college.
- 14. It is important that I understand mathematical concepts

Appendix B – Algebra I Common Core Standards used in this Unit

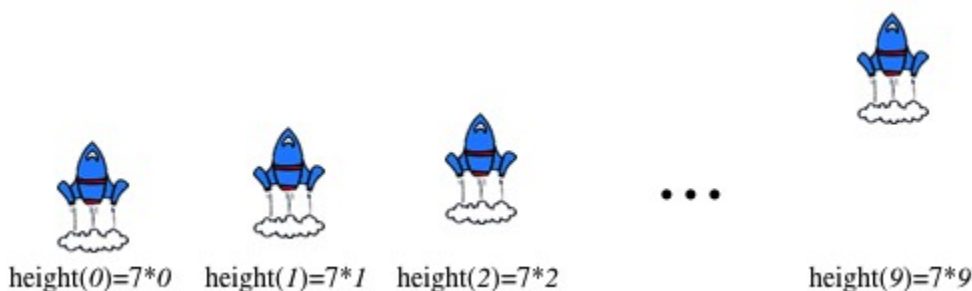
Algebra Standard	Corresponding Unit Lesson
N-Q: Reason quantitatively and use units to solve problems.	Lessons 1-10
6.NS.5-8: Apply and extend previous understandings of numbers to the system of rational numbers.	Lessons 1-10
7.EE.1-4: The student uses numerical and algebraic expressions and equations to solve real-life and mathematical problems.	Lessons 4, 5, 6
8.F.1-3: Define, evaluate, and compare functions.	Lessons 5, 6
A-SSE.1-2: Interpret the structure of expressions	Lessons 3, 4
A-SSE.3-4: Write expressions in equivalent forms to solve problems	Lessons 2,3
A-CED.1-4: Create equations that describe numbers or relationships	Lesson 6
A-REI.1-2: Understand solving equations as a process of reasoning and explain the reasoning	Lessons 3-7
A-REI.3-4: Solve equations and inequalities in one variable	Lessons 8, 9
A-REI.10-12: Represent and solve equations and inequalities graphically	Lesson 7
8.G.6-8: Understand and apply the Pythagorean Theorem	Lesson 9
F-IF.1-3: Understand the concept of a function and use function notation	Lesson 3
F-IF.4-6: Interpret functions that arise in applications in terms of the context	Lessons 4, 5, 6
F-IF.7-9: Analyze functions using different representations	Lessons 3-10
F-BF.1-2: Build a function that models a relationship between two quantities	Lessons 6-10
F-BF.3-4: Build new functions from existing functions	Lessons 6-10

Appendix C – Example of content lesson and a following application lesson

(LESSON TAUGHT IN CLASSROOM)

Algebra Standards 7.EE.1-4, 8.F.1-3, A.CED.1-4, A-REI.1-2, F-IF.4-6, F-BF.1-2, F-BF.3-4

Challenge 6 – Defining your game!



EXAMPLE: A rocket blasts off traveling at 7 meters per second. Write a function called ‘rocket-height’ that takes in the number of seconds that have passed since the rocket took off, and produces the height of the rocket at that time, in meters.

- a. What is the rocket’s height after 0 seconds? _____
 - b. What is the rocket’s height after 1 second? _____
 - c. What is the rocket’s height after 2 seconds? _____
 - d. What is the rocket’s height after 3 seconds? _____
 - e. What is the rocket’s height after 4 seconds? _____
 - f. What is the rocket’s height after 5 seconds? _____
 - g. Mathematically, how can we compute the rocket’s height after ‘x’ seconds?

 - h. What is the *Name* of this function? _____
 - i. What is the *Domain* of this function? _____
 - j. What is the *Range* of this function? _____
 - k. Based on parts (h) through (j) above, the *contract* for this function is:

-

In our programming language, variables can be defined. Here are a few examples:

```
(define x 4)
(define y (+ 4 9))
(define z (* x 2))
```

○ Using the definition to the left, what does x equal? _____
 ○ Using the definition to the left, what will y equal? _____
 ○ Using the definition to the left, what will z equal? _____

1. Convert the following Code definitions into Algebra equations. The first 2 are done for you.

Racket Code	Algebra
(define x 10)	$x = 10$
(define y (* x 2))	$y = x \times 2$
(define z (+ x y))	
(define age 14)	
(define months (* age 12))	
(define days (* months 30))	

2. Convert the following Algebra equations into Code definitions.

Algebra	Code definitions
◦ $dollars = 16.50$	
◦ $feet = 2 \times 3$	
◦ $inches = feet \times 12$	

3. To give more information than is given in a *contract*, programmers write **Purpose Statements**, which are simple sentences that explain what a function does. Often the purpose statement is just a re-statement of the word problem! We'll try a purpose statement for 'rocket-height' on the next page...

Word Problem #1: Complete the contract, purpose statement, 2 examples, and definition for:

Word Problem: rocket-height

Directions: A rocket blasts off, traveling at 7 meters per second. Write a function called 'rocket-height' that takes in the number of seconds that have passed since the rocket took off, and which produces the height of the rocket at that time.

Contract and Purpose Statement

Every contract has three parts ...

f _____ \rightarrow _____
 function name domain range
 f _____
 what does the function do?

Examples

Write some examples of your function in action...

(EXAMPLE _____)
 function name input(s) what the function produces
 (EXAMPLE _____)
 function name input(s) what the function produces

remember to circle the change-able part of the EXAMPLES, then come up with the variable name

Definition

Write the definition, giving variable names to all your input values...

(define _____)
 function name variables

 what the function does with those variables

Word Problem #2: Complete the contract, purpose statement, 2 examples, and definition for:

Word Problem: red-square

Directions: Use the Design Recipe to write a function called 'red-square', which takes in one number (the size of the square) and outputs a solid, red square.

Contract and Purpose Statement

Every contract has three parts ...

;	:	→
function name	domain	range
;		
what does the function do?		

Examples

Write some examples of your function in action...

(EXAMPLE	function name	input(s)	what the function produces)
(EXAMPLE	function name	input(s)	what the function produces)

remember to circle the change-able part of the EXAMPLES, then come up with the variable name

Definition

Write the definition, giving variable names to all your input values...

(define	function name	variables)
what the function does with those variables				

Word Problem #3: Complete the contract, purpose statement, 2 examples, and definition for:

Word Problem: yard-area

Directions: Use the Design Recipe to write a function called 'yard-area', which takes in the width and length of a backyard and returns the area of the yard. (don't forget: Area = length * width)

Contract and Purpose Statement

Every contract has three parts ...

```

; _____ : _____ → _____
  function name          domain          range

;
  _____
  what does the function do?
  
```

Examples

Write some examples of your function in action...

```

(EXAMPLE _____ )
      function name    input(s)          what the function produces

(EXAMPLE _____ )
      function name    input(s)          what the function produces
  
```

remember to circle the change-able part of the EXAMPLES, then come up with the variable name

Definition

Write the definition, giving variable names to all your input values...

```

(define _____ )
      function name    variables

      _____
      what the function does with those variables
  
```

(LESSON CONTINUED IN COMPUTER LAB)

Using the code and your knowledge from your word problems from rocket-height, red-square, and yard-area, answer (a)-(d) by showing your code works

- a. In the Interactions window, type `(rocket-height 20)` to find the height of the rocket after 20 seconds.
- b. Use the rocket-height definition to find how high the rocket will be after 5 *minutes*.
- c. Type correct code into the Interactions window that shows proper use of the function red-square to create a 55 by 55 red square.
- d. Type correct code into the Interactions window that shows proper use of the function 'yard-area'.
For example, try `(yard-area 10 15)` in the Interactions window to see if the area is calculated.

Once you have correct Design Recipes for Word problems 1 through 3, evaluate the following in the Interactions window: `(rocket-height 44)` `(red-square 120)` and `(yard-area 35 20)`

Ask your teacher to check your work from Word Problems 1 through 3: _____

It's time to write the code that will get your Danger and Target moving in your videogame!

Since the Danger can only move to the left, will the x-coordinate or y-coordinate be affected? In what way? _____

Word Problem: update-danger

Directions: Use the Design Recipe to write a function called 'update-danger', which takes in the Danger's x-coordinate and produces the next x-coordinate, which is **50 pixels to the left**.

Contract and Purpose Statement

Every contract has three parts ...

;	:	→	
function name	domain		range
;			
what does the function do?			

Examples

Write some examples of your function in action...

(EXAMPLE)
	function name	input(s)	what the function produces	
(EXAMPLE)
	function name	input(s)	what the function produces	

remember to circle the change-able part of the EXAMPLES, then come up with the variable name

Definition

Write the definition, giving variable names to all your input values...

(define)
	function name	variables		
what the function does with those variables				

- a. Enter the new code you just created and then click "Insert"
- b. On the line right below the contract, type the purpose statement, beginning with a semi-colon. The text should be orange. Click "Run" so the computer will read your code.
- c. Evaluate (update-danger 50) in the Interactions window to make sure that the output given is 50 less. Then evaluate (update-danger 120) to check that it works again.

Since the Target can only move to the right, will the x-coordinate or y-coordinate be affected? In what way? _____

Word Problem: update-target

Directions: Use the Design Recipe to write a function called ‘update-target’, which takes in the Target’s x-coordinate and produces the next x-coordinate, which is **50 pixels to the right**.

Contract and Purpose Statement

Every contract has three parts ...

```

; _____ : _____ → _____
  function name          domain          range

;
  _____
  what does the function do?
  
```

Examples

Write some examples of your function in action...

```

(EXAMPLE _____ )
  function name    input(s)          what the function produces

(EXAMPLE _____ )
  function name    input(s)          what the function produces
  
```

remember to circle the change-able part of the EXAMPLES, then come up with the variable name

Definition

Write the definition, giving variable names to all your input values...

```

(define _____ )
  function name    variables

  _____
  what the function does with those variables
  
```

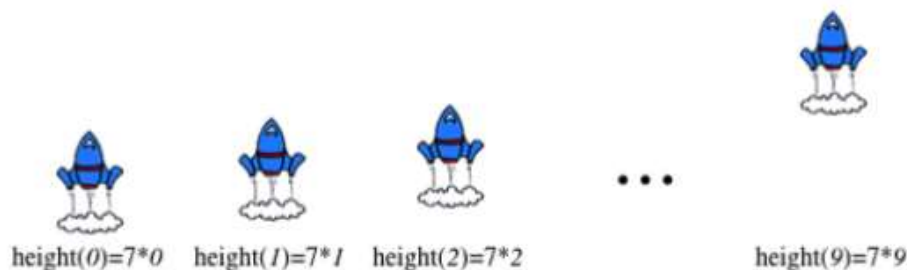
- Enter the new code you just created and then click “Insert”.
- On the line right below the contract, type the purpose statement, beginning with a semi-colon. The text should be orange. Click “Run” so the computer will read your code.
- Evaluate (update-target 50) in the Interactions window to make sure that the output given is 50 more. Then evaluate (update-target 120) to check that it works again.
- When you are ready, have your teacher check your code in the “<Last Name> videogame” file. You’ll know you are ready when you click “Run” and see your Danger and Target fly across the screen!

Teacher initials: _____

SOLUTIONS

Name: _____ Date: _____

Challenge 6 – Defining your game!



EXAMPLE: A rocket blasts off traveling at 7 meters per second. Write a function called 'rocket-height' that takes in the number of seconds that have passed since the rocket took off, and produces the height of the rocket at that time, in meters.

- a. What is the rocket's height after 0 seconds? $0(7) = 0m$
- b. What is the rocket's height after 1 second? $1(7) = 7m$
- c. What is the rocket's height after 2 seconds? $2(7) = 14m$
- d. What is the rocket's height after 3 seconds? $3(7) = 21m$
- e. What is the rocket's height after 4 seconds? $4(7) = 28m$
- f. What is the rocket's height after 5 seconds? $5(7) = 35m$
- g. Mathematically, how can we compute the rocket's height after 'x' seconds? $\text{height}(x) = 7 \cdot x$
- h. What is the **Name** of this function? rocket-height
- i. What is the **Domain** of this function? number (seconds)
- j. What is the **Range** of this function? number (height)
- k. Based on parts (h) through (j) above, the *contract* for this function is:

; rocket-height : number \rightarrow number

In our programming language, variables can be defined. Here are a few examples:

```
(define x 4)
```

```
(define y (+ 4 9))
```

```
(define z (* x 2))
```

○ Using the definition to the left, what does x equal?

$$x = 4$$

○ Using the definition to the left, what will y equal?

$$y = 4 + 9 = 13$$

○ Using the definition to the left, what will z equal?

$$z = x \cdot 2 = 4 \cdot 2 = 8$$

- Convert the following Code definitions into Algebra equations. The first 2 are done for you.

Racket Code	Algebra
<code>(define x 10)</code>	$x = 10$
<code>(define y (* x 2))</code>	$y = x \cdot 2$
<code>(define z (+ x y))</code>	$z = x + y$
<code>(define age 14)</code>	$\text{age} = 14$
<code>(define months (* age 12))</code>	$\text{months} = \text{age} \cdot 12$
<code>(define days (* months 30))</code>	$\text{days} = \text{months} \cdot 30$

- Convert the following Algebra equations into Code definitions.

Algebra	Code definitions
○ $\text{dollars} = 16.50$	<code>(define dollars 16.50)</code>
○ $\text{feet} = 2 \times 3$	<code>(define feet (* 2 3))</code>
○ $\text{inches} = \text{feet} \times 12$	<code>(define inches (* feet 12))</code>

- To give more information than is given in a *contract*, programmers write Purpose Statements, which are simple sentences that explain what a function does. Often the purpose statement is just a re-statement of the word problem! We'll try a purpose statement for 'rocket-height' on the next page...

Word Problem #1: Complete the contract, purpose statement, 2 examples, and definition for:

Word Problem: rocket-height

Directions: A rocket blasts off, traveling at 7 meters per second. Write a function called 'rocket-height' that takes in the number of seconds that have passed since the rocket took off, and which produces the height of the rocket at that time.

Contract and Purpose Statement

Every contract has three parts ...

: rocket-height : Number → Number
function name domain range

: give you height of rocket after some time
what does the function do?

Examples

Write some examples of your function in action.

(EXAMPLE (rocket-height 300) (* 300 7))
function name input what the function produces

(EXAMPLE (rocket-height 24) (* 24 7))
function name input what the function produces
sec sec

remember to circle the change-able part of the EXAMPLES, then come up with the variable name

Definition

Write the definition, giving variable names to all your input values...

(define (rocket-height sec)
function name variables
(* sec 7))
what the function does with those variables

Word Problem #2: Complete the contract, purpose statement, 2 examples, and definition for:

Word Problem: red-square

Directions: Use the Design Recipe to write a function called 'red-square', which takes in one number (the size of the square) and outputs a solid, red square.

Contract and Purpose Statement

Every contract has three parts ...

; RS : number → image
function name domain range

; take any size + produce a red square
what does the function do?

Examples

Write some examples of your function in action...

(EXAMPLE (RS 10) (square 10 "solid" "red"))
function name input() what the function produces

(EXAMPLE (RS 50) (square 50 "solid" "red"))
function name input() what the function produces
size size

remember to circle the change-able part of the EXAMPLES, then come up with the variable name

Definition

Write the definition, giving variable names to all your input values...

(define (RS size)
function name variables
(square size "solid" "red"))
what the function does with those variables

Word Problem #3: Complete the contract, purpose statement, 2 examples, and definition for:

Word Problem: yard-area

Directions: Use the Design Recipe to write a function called 'yard-area', which takes in the width and length of a backyard and returns the area of the yard. (don't forget: Area = length * width)

Contract and Purpose Statement

Every contract has three parts ...

: yard-area : number number → number
function name domain range

: gives us the area
what does the function do?

Examples

Write some examples of your function in action...

(EXAMPLE (yard-area 5 8) (* 5 8))
function name input(s) what the function produces

(EXAMPLE (yard-area 10 12) (* 10 12))
function name input(s) what the function produces
L W L W

remember to circle the change-able part of the EXAMPLES, then come up with the variable name

Definition

Write the definition, giving variable names to all your input values...

(define (yard-area L W)
function name variables
(* L W)
what the function does with those variables)

It's time to **write the code** that will **get your Danger and Target moving** in your videogame!

Since the Danger can only move to the left, will the x-coordinate or y-coordinate be affected? In what way? X-coordinate decreases

Word Problem: update-danger

Directions: Use the Design Recipe to write a function called 'update-danger', which takes in the Danger's x-coordinate and produces the next x-coordinate, which is **50 pixels to the left**.

Contract and Purpose Statement

Every contract has three parts ...

: update-danger number → number
function name domain range

: takes the x-coord and subtracts 50
what does the function do?

Examples

Write some examples of your function in action...

(EXAMPLE (update-danger 40) (- 40 50))
function name input(s) what the function produces

(EXAMPLE (update-danger 20) (- 20 50))
function name input(s) what the function produces

remember to circle the change-able part of the EXAMPLES, then come up with the variable name

Definition

Write the definition, giving variable names to all your input values...

(define (update-danger x) (- x 50))
function name variables what the function does with those variables

- Enter the new code you just created and then click "Insert"
- On the line right below the contract, type the purpose statement, beginning with a semi-colon. The text should be orange. Click "Run" so the computer will read your code.
- Evaluate (update-danger 50) in the Interactions window to make sure that the output given is 50 less. Then evaluate (update-danger 120) to check that it works again

Since the Target can only move to the right, will the x-coordinate or y-coordinate be affected? In what way? X-coordinate increases

Word Problem: update-target

Directions: Use the Design Recipe to write a function called 'update-target', which takes in the Target's x-coordinate and produces the next x-coordinate, which is **50 pixels to the right**.

Contract and Purpose Statement

Every contract has three parts ...

; update-target number → number
function name domain range

; takes the x-coord and adds 50
what does the function do?

Examples

Write some examples of your function in action...

(EXAMPLE (update-target 40) (+ 40 50))
function name input(s) what the function produces

(EXAMPLE (update-target 20) (+ 20 50))
function name input(s) what the function produces

remember to circle the change-able part of the EXAMPLES, then come up with the variable name

Definition

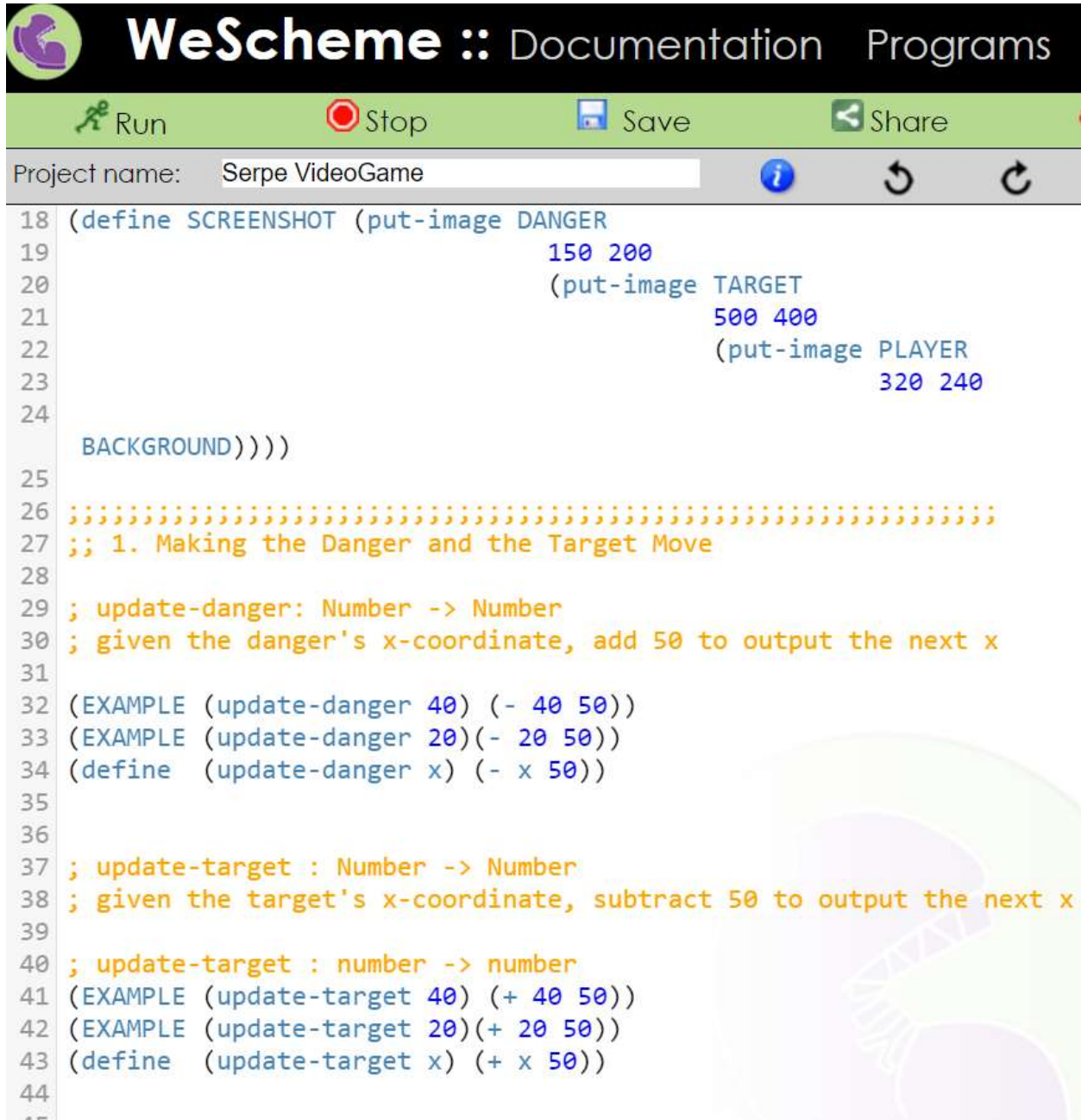
Write the definition, giving variable names to all your input values...

(define (update-target X) (+ X 50))
function name variables what the function does with those variables

- Enter the new code you just created and then click "Insert".
- On the line right below the contract, type the purpose statement, beginning with a semi-colon. The text should be orange. Click "Run" so the computer will read your code.
- Evaluate (update-target 50) in the Interactions window to make sure that the output given is 50 more. Then evaluate (update-target 120) to check that it works again.
- When you are ready, have your teacher check your code in the "<Last Name> videogame" file. You'll know you are ready when you click "Run" and see your Danger and Target fly across the screen!

Teacher initials: _____

In the computer lab, their code should now look like this, and their danger will move across the screen to the left, while their target will move across the screen to the right.



The screenshot shows the WeScheme IDE interface. At the top, there's a header with the WeScheme logo and the text "WeScheme :: Documentation Programs". Below the header is a toolbar with buttons for "Run", "Stop", "Save", and "Share". The "Project name" field is set to "Serpe VideoGame". The code editor displays the following code:

```

18 (define SCRENSHOT (put-image DANGER
19                       150 200
20                       (put-image TARGET
21                           500 400
22                           (put-image PLAYER
23                               320 240
24
25 BACKGROUND))))
26 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
27 ;; 1. Making the Danger and the Target Move
28
29 ; update-danger: Number -> Number
30 ; given the danger's x-coordinate, add 50 to output the next x
31
32 (EXAMPLE (update-danger 40) (- 40 50))
33 (EXAMPLE (update-danger 20)(- 20 50))
34 (define (update-danger x) (- x 50))
35
36
37 ; update-target : Number -> Number
38 ; given the target's x-coordinate, subtract 50 to output the next x
39
40 ; update-target : number -> number
41 (EXAMPLE (update-target 40) (+ 40 50))
42 (EXAMPLE (update-target 20)(+ 20 50))
43 (define (update-target x) (+ x 50))
44
45

```