MERGING EDUCATION, ASSESSMENT, AND ENTERTAINMENT IN MATH GAMES: A CASE STUDY OF FUNCTION FORCE

A Case Study of Function Force

MIKE TREANOR, CHRISTOPHER W. TOTTEN, JOSHUA MCCOY, AND G. TANNER JACKSON

Author Note

This work was partially funded by the Educational Testing Service, and was developed along with Dan Petricca, Kirby Cofino, Gray Leonard, and Gabrielle Cayton-Hodges.

Abstract

Games for education and games for entertainment tend to follow different development paths according to their respective purposes. Entertainment game development follows the traditional best practices of games design while educational game development adds an additional constraining layer of learning objectives. The contribution of this work is a case study of a game, Function Force, that was created with the entertainment and educational design goals given relatively equal weight. In this case study, the development of the game mechanic, puzzle designs, and game level progressions will be explained and connected to relevant educational and curriculum components.

Introduction

The field of educational games has the goal of creating gameplay experiences that impact learning (Dondlinger, 2007). In this field, representing problems and concepts necessary to education is the primary focus (Squire, 2006). Games such as Sim City (Maxis, 2013), Rome: Total War (The Creative Assembly, 2004), and Minecraft (Mojang, 2011) are used in educational settings because the gameplay they provide happens to be consistent with educational objectives (Minnery & Searle, 2014; Nebel, Schneider, & Rey, 2016). Other educational games impart lessons from the humanities through environmental storytelling but lack meaningful interactive game mechanics. Other than notable exceptions (Aylett, Louchart, Dias, Paiva, & Vala, 2005; Bell, 2009; Rowe, Shores, Mott, & Lester, 2011), games that are developed explicitly as educational material that have meaningful interactive mechanics are driven by the educational objective and relegate principled game design to a lower tier of importance.

This leaves open the correlation between following the best practices of game design and educational outcomes. Through merging principles in game design with those of assessment and curriculum objectives, engaging gameplay can be created that is integrated tightly with educational practices. Such practices include: game feel/juice consisting of the visual and auditory feedback produced by the game (Swink, 2008); the balance of the gameplay mechanics and systems; the flow and designed progression of the gameplay experience (Chen, 2007); the design of the levels to achieve objectives for player experience; and the playable models that players interactive with and compose the basic interactivity loop of the game (Wardrip-Fruin, 2009).

In 2011 President Barack Obama called for investment in educational technology, challenging developers to create educational software "as compelling as the best video game." (DeLoura, 2017). In response, the White House Office of Science and Technology Policy gathered commercial and academic game developers for the White House Education Game Jam in 2014. Here, we first prototyped Function Force. a "shoot 'em up" game meant to intuitively teach mathematical functional transformations. The game was redeveloped in 2017 in collaboration with the Educational Testing Service (ETS) as an online game-based assessment. In this instance, the game targets students who have advanced past simple mathematical operations at roughly the 8th grade level.

Function Force was designed around the hypotheses that there is value in making the educational content directly playable and that entertainment and educational goals could be met by transforming preexisting genre tropes into emergent teaching mechanisms. Particularly, Function Force makes use of the tropes of the shoot 'em up (or shmup) genre to give players the experience of playing with function transformations in real time to achieve game goals. We believe that the successful completion of these goals signals a progression in the education of the player.

The new purpose of the game – quantitative measurement and testing – demanded a rethinking of Function Force: gameplay had to follow established understandings of learning progression in order to produce scientific results from user studies (Arieli-Attali, Wylie, & Bauer, 2012). While the original version of Function Force aimed to allow interaction with functions in the "most fun" way, in a collaborative effort with ETS the game was redesigned to emphasize more of the core mathematical operations of function transformations and how to incrementally introduce new gameplay functionality according to established progressions. With the new version, gameplay elements needed to be revealed gradually in order to produce discrete educational lessons that build upon each other. The new game design had to confront a common issue that most educational games face: how can a game both be fun and educational, while adhering to an empirically validated curriculum, such as the Common Core Grade 6-8 Mathematics standards (McREL International, 2014).

In this paper, we contribute a case study of the game Function Force that describes the deliberate merger of game design and educational practices. With Function Force as our example, we argue that making the educational content playable, rather than layered, is a fruitful way to developing education games. We believe that Function Force does not merely utilize rewards to incentivize students during the often-rote educational sections of the game, but it rather brings about educational moments in a fun, challenging, and dynamic manner that is directly part of the gameplay. In this case study, the development of the game mechanic, puzzle designs, and game level progressions will be explained and connected to relevant educational and curriculum components.

Educational Goals

Function Force primarily aims to teach students about linear functions (specifically the concept of slope and y-intercept). Linear functions are often the first functions that students are introduced to, and rather than presenting these concepts directly, or through problem sets, Function Force gives students the opportunity to gain an intuitive understanding of these concepts through play as they demonstrate their understanding while achieving gameplay goals. As demonstrated by Squire in his work on the efficacy of games for educational purposes (Squire, 2004), the idea is that after gaining an intuitive understanding of the concepts through play, students will be more able to grasp the concepts when presented in more traditional educational environments (i.e., preparedness for future learning).

The main mathematical concept that Function Force is centered around is the equation . In that equation, the constant 'm' describes the rate of change for a result (y) as x increases. This is often called the function's 'slope'. The constant 'b' describes the 'y-intercept' of the function. The y-intercept is the resulting value of the function when the input, x, is zero. That is, this is the point from which the function begins sloping upward (or downward) as the input, x, changes.

A hypothesis of this project is that these concepts are much better understood when used in intrinsically motivated scenarios (or when a player is in a flow state, as described in some game design communities) (Salen and Zimmerman, 2003, p 336). Rather than presenting a student with these concepts abstractly, Function Force presents these concepts as tools to complete goals in an entertaining 'shmup' game.

The following section will describe how we created game mechanics, designed levels, and sequenced those levels to adhere to a progression that demonstrates that, when completed, players will have gained an understanding of how the slope and y-intercept operate in the function.

Case Study: The Design of Function Force

Function Force is a shmup game that puts the player in control of a futuristic spaceship that destroys robotic dinosaurs. Some popular shmups include Space Invaders, R-Type, and Ikaruga. Unique to Function Force, however, is that players fire upon enemies with a laser whose trajectory is determined by modifying an equation displayed on the screen.

The Ship and Weapons

Gameplay in Function Force primarily involves controlling the vertical position of a ship moving at a constant rate from left to right, avoiding and destroying enemies, and collecting and configuring power ups to change the direction of the ship's weapons.

In Function Force, the weapon plays the critical role of firing projectiles that travel along a path set by the linear mathematical function: $y = mx + b$. This function is displayed on the bottom right of the screen, and at the beginning of the game, the function displays y = x (i.e. y = 1x). In Function Force, the x represents the distance in front of the ship, and y is the distance above and below (i.e. a traditional two-dimensional coordinate plane with the ship at the origin). As the player moves forward through the first level, the player encounters two types of power ups: the plus (+) and minus (-). Each power up affects the 'm' value in the $y = mx + b$ function. In other words, when the player first starts, the equation for the laser is y=0x and shooting results in a straight horizontal line from the front of the ship (since the ship is the origin and the y-intercept is 0). If the player collects a plus (+) power up, the function displayed becomes y = 1x, and the weapon will now shoot aiming upwards at 45 degrees. Otherwise, if the player were to collect the minus (-) power up, they would be aiming downward at 45 degrees. While a player need not notice the displayed function in the bottom-right corner of the screen at the start of the game, as they encounter puzzles they will need to pay closer attention to the function's m, or slope, value.

Puzzles as Math Problems

Where the spaceship is the player's primary mechanism for interacting with the linear function, the game utilizes puzzles to test the player's mastery of the ship. If one envisions Function Force as an interactive math test, puzzles are individual problems for assessing the player's mastery of transforming the $y = mx + b$ function. In internal game design notes, these puzzles were also referred to as "static sections" of gameplay, as the game stops all forward movement until the player completes the puzzle. In these puzzles, the player must shoot a switch with their weapon to open a gate and continue forward progress through the game. The player's challenge is to determine the proper slope (m value) that must be used to aim their weapon towards the switch and adjust their weapon with power ups to activate the switch. Figure 1 shows an early puzzle that exemplifies this dynamic.

Figure 1: In an early puzzle, the player sees a switch located in the sloped tunnel above the player's ship. To activate the switch they must adjust the m value of their ship's weapon. The scene provides all the tools necessary to solve the puzzle and understand when the problem has been completed.

Within Function Force, these puzzles were designed as "scenes" and serve as "the most basic unit of pacing in [the] game" (Anthropy & Clark, 2014, p. 40). As per Anthropy et al., the various amount of screen-space dedicated to a scene is open to interpretation since they "might use completely different sizes and shapes for scenes." (Anthropy & Clark, 2014, p. 40) Scenes can be used to present players with gameplay problems to solve and give them the required means to do so within a small area of level space. These scenes, are there to introduce and develop the actions available to a player in a game

(Anthropy & Clark, 2014, p. 41), which in Function Force entails modifying the $y = mx + b$ function in various ways to progress within the game.

It was decided to have each static puzzle scene occupy a single screen's worth of gameplay space. Within this space would be the switch (or switches in later levels) that players had to hit with their spaceship's laser, level geometry that would stop the weapon, a consistent supply of power ups for adjusting the weapon, and the exit door. In this way, each assessment scene provides players with everything needed to solve them, presented in a way that maximizes usability and gameplay balance. Previous studies of level design and usability have established that a single screen is an effective space for introducing complex gameplay problems (Totten, 2014, p. 174) and the tools for solving them.

Satellites add Complexity to Gameplay and Math Concepts

At the end of level two, the player collects a power up which we call the 'satellite'. The satellite is an additional drone ship that follows the vertical movement of the player and shoots its own weapon as the player presses the fire button. The satellite's vertical offset from the player's position, and the slope of the weapon it shoots, is set by an interface where the player directly enters the m/slope, and b/y-intercept (figure 7). The origin for the satellite's function is the player's main ship, and thus, the player can adjust the relative distance of the satellite (above/below the main ship) by selecting from the available "b" values (-4,-2,0,2,4). In level three, the player collects a second satellite which they can also directly manipulate the offset and the slope of its weapon through an interface.

These satellites provide new and interesting gameplay affordances and also allow for more complex puzzles that involve hitting multiple switches simultaneously with unique combinations of $y = mx + b$ equations.

Level Layout as Math Lessons

If Function Force's static puzzle scenes represent individual mastery problems, then its levels are quiz sections that assess overall mastery of a math concept. Rogers describes levels as a "confined area of play activity" (Rogers, 2014) and further comments that levels may be distinguished from one another via distinct settings, characters, locations, gameplay mechanics, artwork, or music (Rogers, 2016). In Function Force, levels are constructed by placing several scenes together in sequence. In addition to the static puzzle scenes, levels also contain "auto-scrolling sections" where the spaceship flies forward at a fixed pace for several screens worth of space. During the auto-scroll sections, the player may experiment with modifying the *m* value of the weapon by collecting power ups. Enemy ships also appear in these sections, making them feel like classic shmups.

Alternating auto-scrolling sections and static puzzle scenes within levels serves several educational purposes. First, auto-scrolling sections early in the game give players opportunities to acquaint themselves with the spaceship's mechanics: how it moves, how to fire the weapon, and how power-ups affect the slope of the weapon. Secondly, they provide a variation to the gameplay and pacing (interspersed between puzzle sections) by including more traditional shmup video game experience. Shmups are a genre known for highly satisfying action that maintain player interest over long periods through "game feel" or "juice" – particle effects, sound, and visual elements that provide satisfying feedback to players (Keogh, 2017; Swink, 2008). Lastly, they allow designers to teach gameplay by designing small scenes that force player decisions. Unlike static puzzle scenes, auto-scrolling scenes force the player to steer their ship away from level geometry to continue forward progress. This type of forced decision is an effective way to teach gameplay mechanics and integrate educationally relevant behaviors. For example, in one early scene, a fork in the auto-scrolling tunnel forces the spaceship to pick up either a plus or minus power up. In this way, the player is led towards an icon they may otherwise avoid and can immediately observe the effect that the power up has on both the linear equation on the screen and how their weapon fires (figure 2.). In other areas the level geometry creates sloped tunnels that require the player to grab the correct power-ups to adjust their laser slope to match that of the tunnel space.

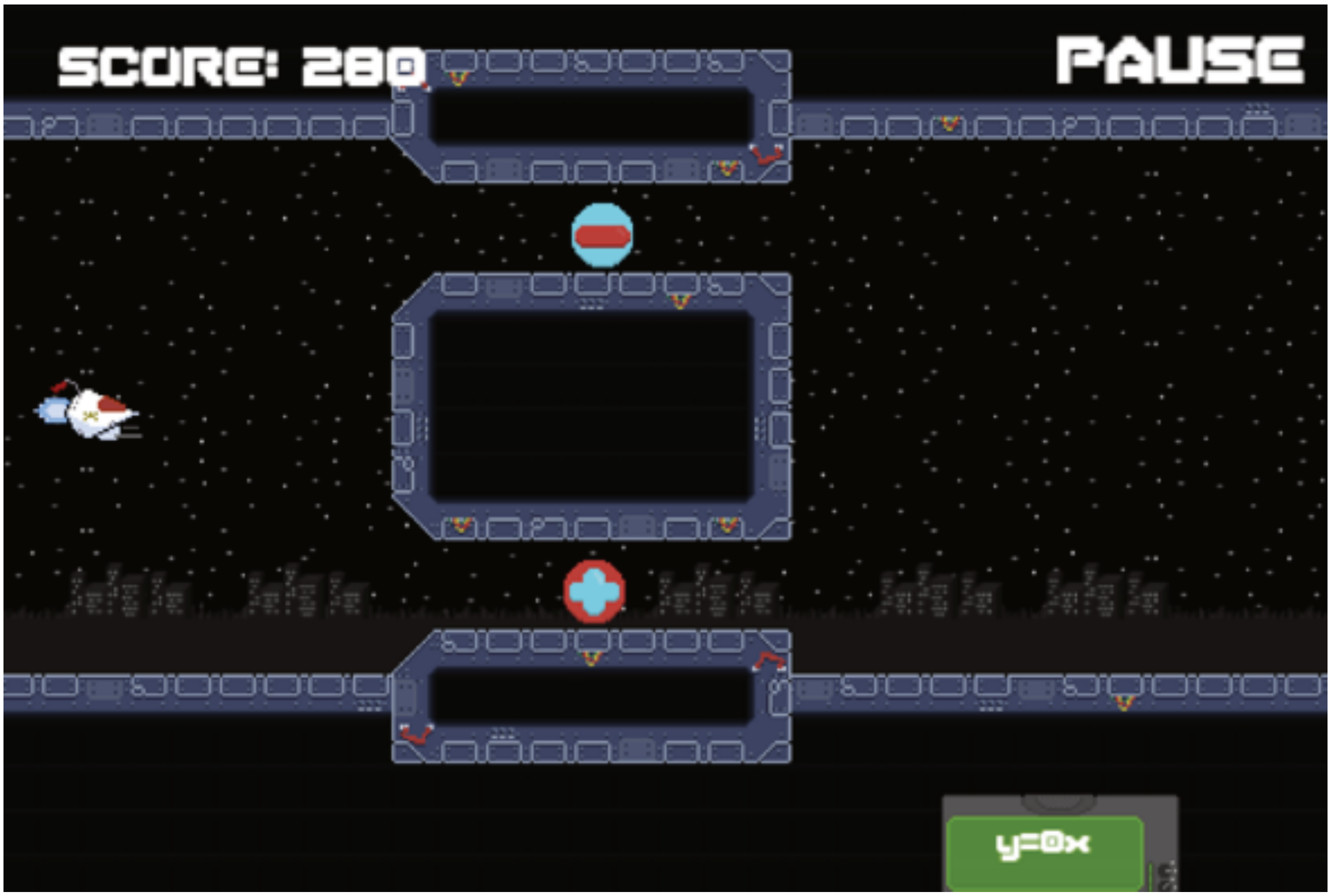

Figure 2: In this early gameplay scene, located in an auto-scrolling section of the level, the player is forced into one of two tunnels where they will collect one of the game's power ups. In this way, the game teaches the player about the function of power ups and how the linear equation is modified.

Another important function of structuring a series of related scenes in a level is enforcing existing learning progressions into gameplay. For gameplay designers, a common learning progression for teaching game mechanics to players is "kishotenketsu" (Brown, 2015), a term borrowed from Japanese manga art. *Super Mario 3D Land* director Koichi Hayashida describes kishotenketsu progression thus: "you introduce a concept, and then in the next panel you develop the idea a little bit more; in the third panel there's something of a change-up, and then in the fourth panel you have your conclusion." (Nutt, 2012) In level design terms, this means introducing a gameplay mechanic in a safe space with few obstacles, then presenting players with a more complex version of the mechanic, introducing the mechanic again in a new context (alongside another different mechanic, for example), then having a final test of the player's mastery of the mechanic. These tests are enforced by "skill gates", obstacles that stop a player from progressing forward until they perform an in-game action or actions that allow them to pass the obstacle. (Kremers, 2009)

These gameplay progressions are mirrored in common learning standards for mathematics (McREL International, 2014) and learning progressions (Arieli-Attali, Wylie, & Bauer, 2012). The Common Core standards for mathematics in particular use "subject clusters" (McREL International, 2014), which are sets of concepts (or "steps") that students must master to make it through the standards. Function Force uses these subject clusters as outlines for what the content of each level should be. Each individual step of the cluster is therefore a game level in which the game design teaching method of kishotenketsu is applied. This means that for Function Force, and games that combine traditional gameplay and educational content, level design is the instrument of administering and assessing educational content. In Function Force specifically, levels utilize kishotenketsu to introduce a mathematical concept with a simple mechanic (e.g., vertical movement and sloped laser to hit easily encountered enemies), then present a slightly more difficult version (e.g., acquire specific slope value to hit switches), a version that combines the concept with other previous concepts (e.g., add satellite drone with manipulable equations), then more difficult final test of the player's mastery of the concept (e.g., two drones with multiple switches, some which might be moving). Between the static scenes, auto-scroll shmup scenes are placed to provide more fast-paced gameplay and allow players to explore mechanics such as collecting power ups or operating new weapons (such as the satellites).

Educational Challenge Progression

As discussed previously, the Function Force team looked to educational standards such as Common Core or established Learning Progressions (LPs) to map out how game mechanics would progress over a series of levels (Arieli-Attali, Wylie, & Bauer, 2012). Common Core utilizes "clusters", sets of lessons that lead up to a learning goal, to embody topics. Within these clusters are "steps", which represent individual competencies or lessons that lead to this learning goal. Specifically, Function Force's levels follow the cluster for teaching students to "analyze and solve linear equations and pairs of linear equations" (McREL International, 2014). Again, if individual puzzles are math problems and levels are quiz sections that contain these problems, Function Force as a game with a series of levels is a lesson plan about linear functions.

The primary content of Function Force assesses players' understanding of single-variable linear equations and working with multiple linear equations at one time. Examples from the game shown in previous figures were selected from early levels that utilize single-variable linear equations to describe how Function Force levels are generally constructed. In thinking of a lesson plan as a series of game levels, the designers saw an opportunity to see how education lessons could map alongside the game designer's concept of "flow", an "optimal experience" (Csikszentmihalyi, 1990) state that game players reach when they are fully engrossed in the game. Salen and Zimmerman describe flow as a state of "focused and engaged happiness" (2004, p. 336) that occurs when the challenges of a game balance with the skills players have built while playing the game (Salen and Zimmerman, p. 351). In game development practice, flow is a useful concept for mapping out how balanced a game is over a set of challenges or even over the course of the entire game. By testing and adjusting how player skill and challenges are balanced, the designer can keep players engaged over time.

Function Force's first level represents both the game's low-level flow state: the player is inexperienced with the game and challenges are simple, and the first step in the linear functions cluster: solving linear equations with one variable. The player flies through an enemy space station with generously proportioned spaces that lets them experiment with the space ship's movement capabilities (Totten, 2014, p. 120). Slowly moving enemies also appear in this early stage so players can practice firing the ship's weapon with the rewarding feedback of seeing the "robo-dino" enemies explode. Eventually, the level presents a forced decision that leads the player to collect a power up that changes the slope of their weapon, as shown previously in figure 2. Then the player plays a similar large space with robo-dinos, but with a newly-modified laser. They learn how the power up has changed their weapon and must modify their gameplay strategy to account for this. The player eventually encounters a puzzle that tests their understanding of this dynamic, as shown in figure 1, where the game assesses them on their mastery of the changing slope concept. The level continues in this way with increasingly difficult shmup sections and puzzle scenes testing the player's understanding of the slope mechanic (figure 3).

Figure 3: Beyond the initial teaching instances of how to use the game's slope-changing mechanic, Function Force's first level leads the player through several challenges that test their early understanding of how changing the m value changes the slope of the laser.

The second level more directly tests the player's understanding of single linear functions. It does this by utilizing more difficult gameplay design and puzzles with more obscure solutions than in the previous level. In this level, the player flies through an asteroid with narrow tunnels filled with enemies and must grab the appropriate power-ups to match the slope of their laser to the slope of the tunnel (figure 4).

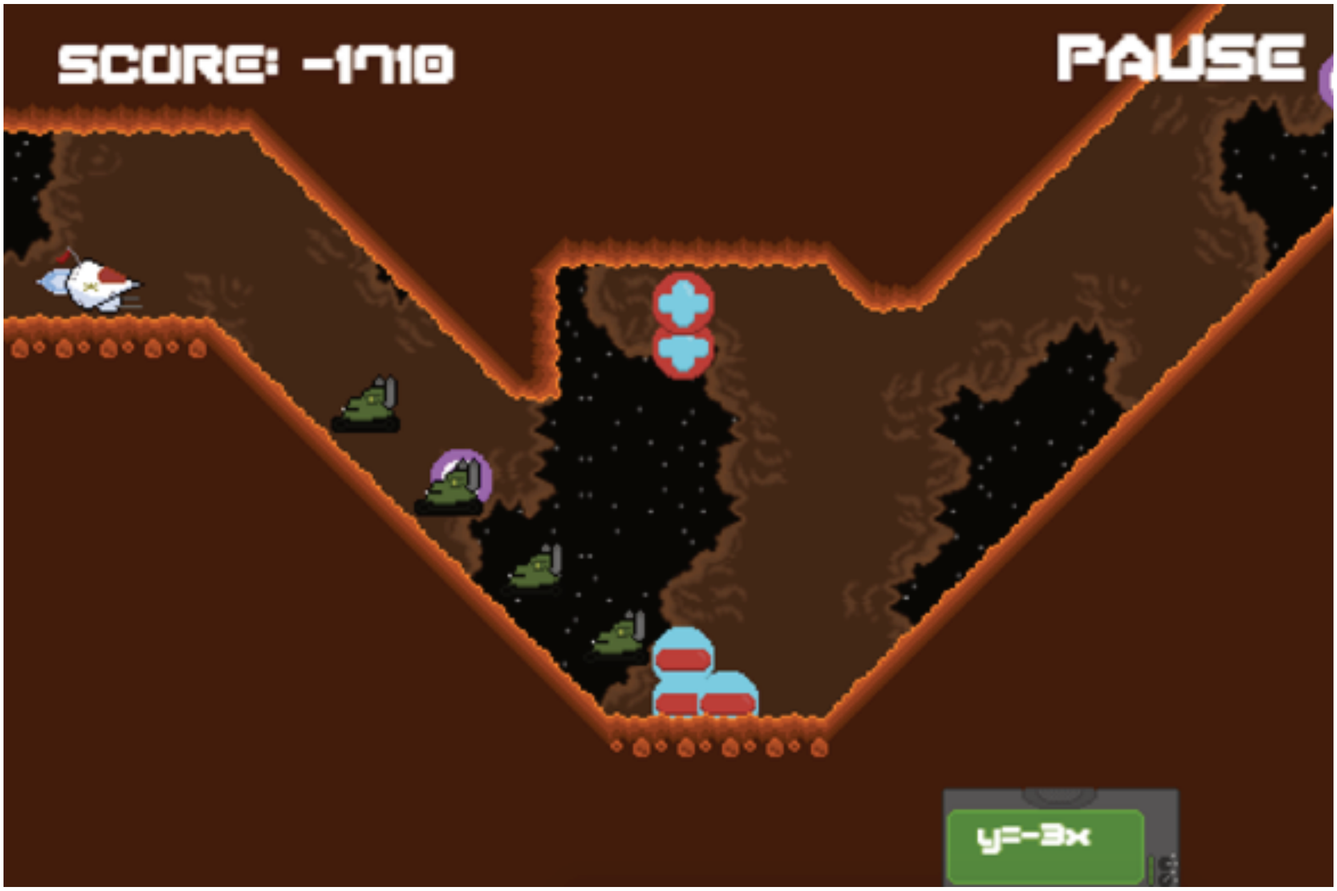

Figure 4: The second level of the game features narrow tunnels that test the player's ability to set their weapon's slope to the right setting.

These narrow tunnels limit player movement and put them into the path of enemies (Totten, 2014, p. 119). This section tests the player's mastery of changing their weapon's slope, as matching the slope value to the slope value of the tunnel is required to successfully pass this section. Figure 4 shows an instance where the level forces another decision for the player: using alcoves to avoid enemies also forces the player to pick up power ups that change their weapon's slope in the correct direction.

The static puzzle scenes in level two also increase in difficulty. Where the first level featured simple and direct tunnels between the player and the switch, the second level forces the player to fire the weapon through complicated obstacles that require the player to solve for *m* (figure 5).

In this way, the player must not only think more deeply about the value they will input for *m*, but also about where to put the function's origin (the ship itself.) In a test of both gameplay and mathematic mastery, the second level eventually features puzzles with moving barriers that can both hurt the player and require timing to fire the weapon around (figure 6).

Figure 5: A puzzle from Function Force's second level features a more complicated linear function problem for the player to solve.

Having mastered modifying the *m* value with the power ups, the third level introduces an upgrade to the game's interface that lets the player modify the weapon's slope freely. This allows the player to adjust the weapon any time they wish, eliminating the restriction and challenge of collecting power ups. However, it also removes the guided numeric changes that the power ups provided: the onus is on the player to select the proper value for m. As the level progresses, it introduces new steps in the cluster, specifically those for "analyzing" and "solving for multiple linear equations" (McREL International, 2014). The game does this by introducing satellites, mini-ships that follow the player's own ship (figure 7).

Figure 6: This puzzle tests both gameplay and mathematical mastery by featuring moving platforms that can damage the player. In this way the player must find a solution to the linear function problem and fire the weapon with proper timing.

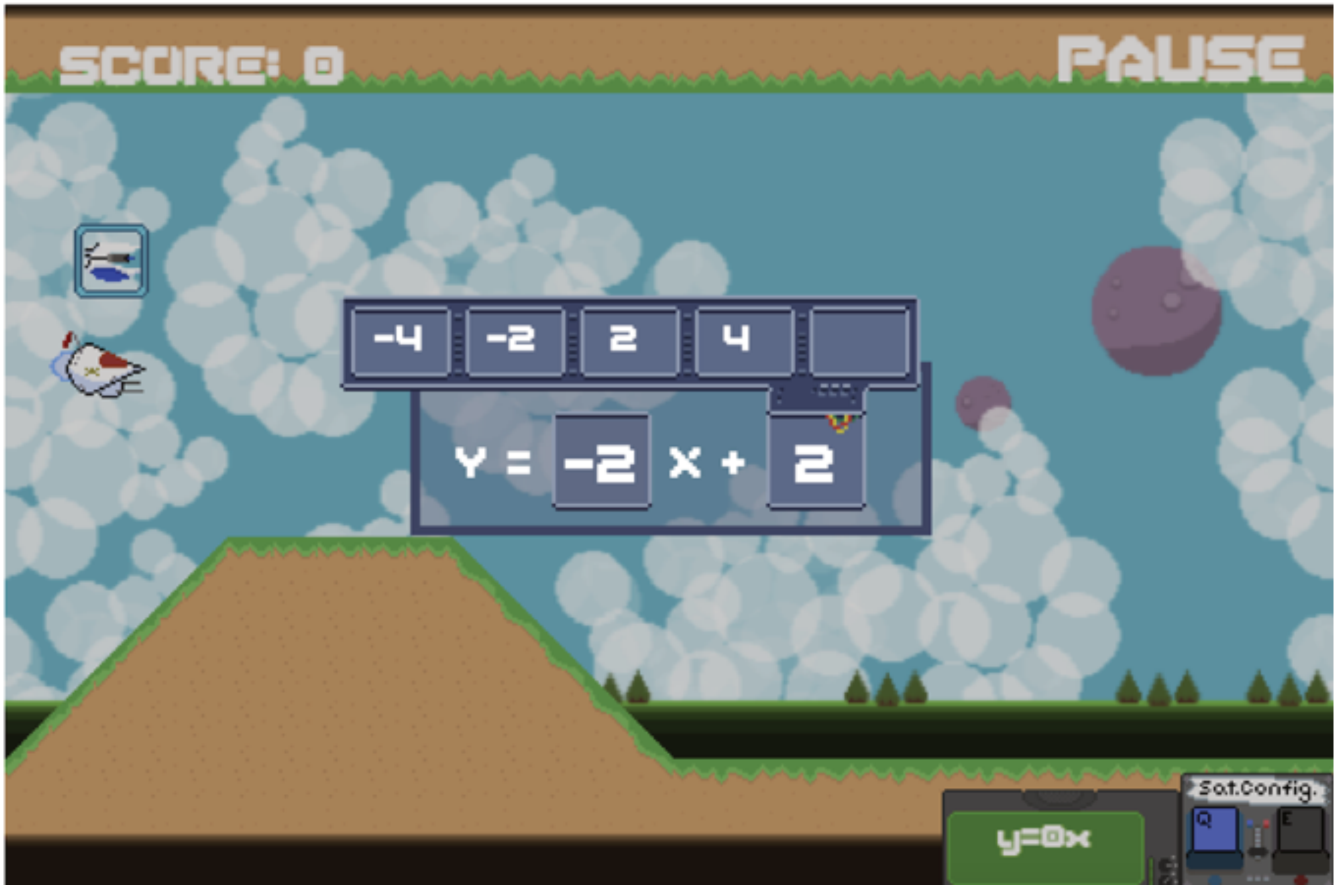

Figure 7: The third level introduces satellites which fire their own weapon along the y = mx + b value. The satellites also include the b value, representing the distance of the satellite from the origin point of the graph (the location of the player's spaceship.) The player is able to directly change each of the satellite's 'm' and 'b' values via the interface above.

The satellites have their own weapon, which like the ship's weapon, fire along the slope represented by the m value. Each satellite (there are eventually two) has a linear function independent of the player spaceship's, meaning that the player can fire up to 3 crisscrossing lasers each with their own slope. The satellites utilize the full $y = mx + b$ equation, where b represents the distance of the satellite on axis y from the origin of the graph, represented in this case by the player's spaceship. This allows for greater gameplay flexibility and greater "game feel" as the player can now fire multiple weapons (figure 8). It also continues the development of the player's flow state: they have more skills in both gameplay and the learning objectives now, so the game's challenges in these areas increase.

Figure 8: The satellites allow the player to fire multiple lasers in an attempt to create more satisfying "game feel.". It also allows the designers to ramp up both mathematical and gameplay challenges, such as introducing more waves of difficult enemies.

Figure 9: Towards the end of the game, the player must manipulate the m and b variables of multiple linear functions and have them reach multiple points at one time.

Towards the end of the third level, the player receives another satellite, further testing their management and ability to utilize multiple linear functions. At the end of the game, the player is able to manipulate both the *m* and *b* values in multiple linear functions and apply those values to puzzles, where they must hit multiple switches at the same time (figure 9). For the puzzle depicted in Figure 9 all three switches must be hit at the same time for the gate to open.

Future Directions and Conclusions

Function Force was developed as both an educational game, aimed at teaching about linear functions, and a testbed for experiments in game-based assessment. Future work is planned that may include experiments comparing Function Force to other digital tasks on linear functions as well as studies that manipulate design features of the game which are hypothesized to impact learning and engagement (e.g., providing feedback through points).

The design of Function Force also affords many more opportunities for educational experiences about function transformation, and the project may be further developed in the future. Other prototypes of the game include sine wave and parabolic function configurations for the ship's weapon. Rather than solely modifying the slope and y-intercept of the weapon, the player could be given the ability to modify the frequency and amplitude, or the coefficients of the quadratic function, while solving puzzles and defeating enemies (figure 10). Furthermore, it is worth exploring the affordances of allowing the player a way to modify the coefficients continuously. We believe that this direct feedback loop, where players quickly see the results of raising and lowering values, will allow for fluid exploration and an intuitive understanding of how various functions operate.

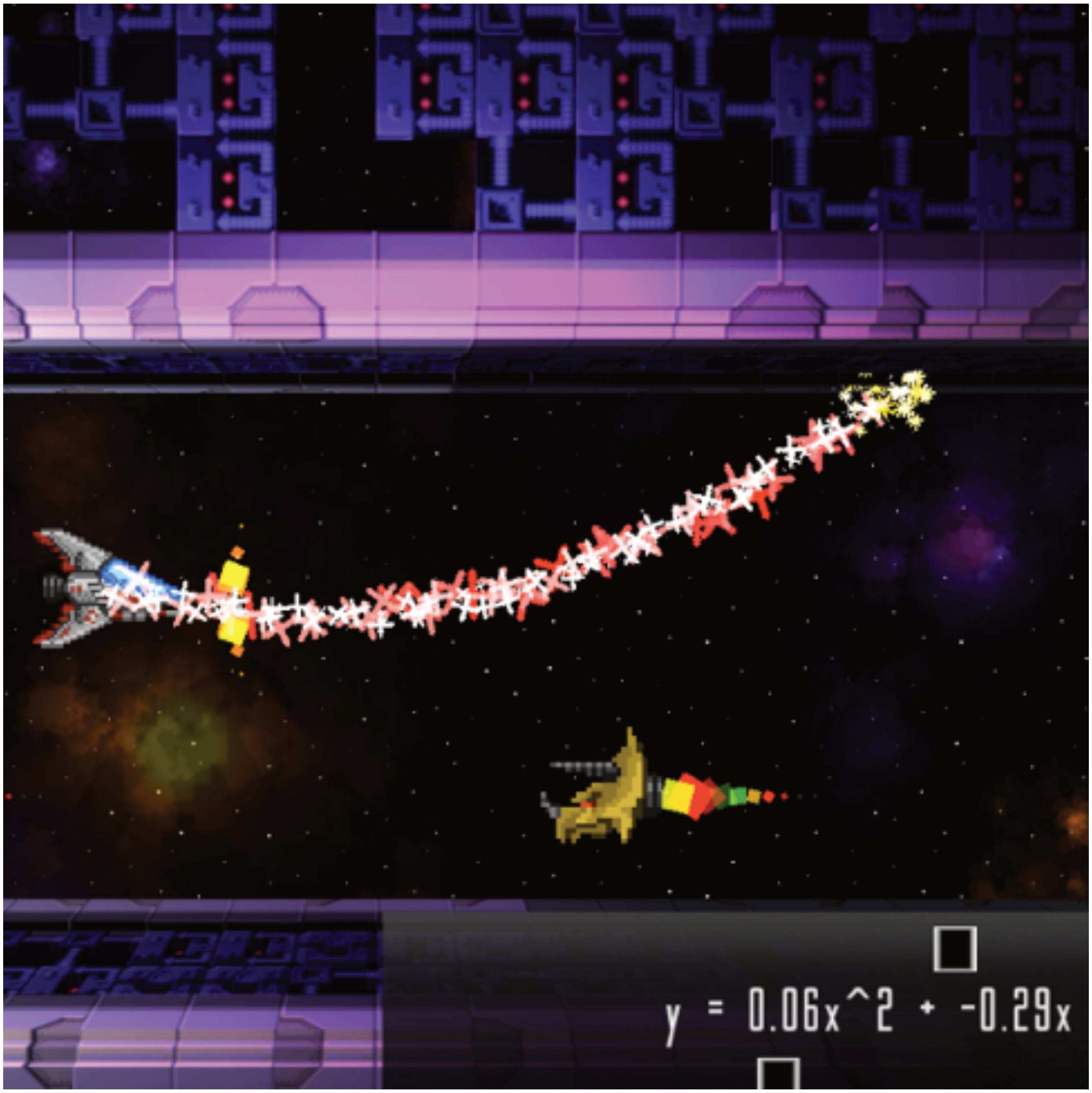

Figure 10: A prototype of Function Force where the player can continuously modify the coefficients via a slider interface. Functions in this version include parabolic and sine waves.

In conclusion, the educational and entertainment design goals were given relatively equal weight in the creation of Function Force. In this case study, the implementation and design of the basic mechanics of the game were explained in terms of the educational and curriculum design they addressed. Gameplay dynamics were explored through puzzle design and puzzles were contextualized through level design which enacts educational practices. While empirical results are forthcoming, we believe that this approach can lead to engaging games that can instill intuitive understandings about important educational material.

References

Arieli-Attali, M., Wylie, E.C., Bauer, M.I. (2012, April). The use of three learning progressions in supporting formative assessment in middle school mathematics. Paper presented at the annual meeting of the American Educational Research Association, Vancouver, Canada.

Aylett, R. S., Louchart, S., Dias, J., Paiva, A., & Vala, M. (2005). FearNot!: An experiment in emergent narrative. In T. Panayiotopoulos, J. Gratch, R. Aylett, D. Ballin, P. Olivier, & T. Rist (Eds.), *Proceedings of the Fifth International Conference on Intelligent Virtual Agents* (Vol. 3661, pp. 305–316). Springer Berlin Heidelberg.

Anthropy, Anna and Naomi Clark. *A Game Design Vocabulary.* Upper Saddle River, NJ: Pearson Educational. 2014. pgs. 40-41.

Bell, D. (2009). Learning from Second Life. *British Journal of Educational Technology*, *40*(3), 515–525. https://doi.org/10.1111/j.1467-8535.2009.00943.x

Brown, Mark. (2015, March 16). *Game Maker's Toolkit: Super Mario 3D World's 4 Step Level Design.* Retrieved from https://www.youtube.com/watch?v=dBmIkEvEBtA.

Chen, J. (2007). Flow in Games (and Everything else). *Commun. ACM*, *50*(4), 31–34. https://doi.org/10.1145/1232743.1232769

Csikszentmihalyi, Mihaly. *Flow: The Psychology of Optimal Experience.* New York: Harper Collins. 1990.

Dondlinger, M. J. (2007). Educational Video Game Design: A Review of the Literature, *4*(1), 12.

DeLoura, Mark. "The White House Education Game Jam." The Obama White House Archive. Created October 4, 2014. https://obamawhitehouse.archives.gov/blog/2014/10/06/white-house-education-game-jam. Accessed May 19, 2017.

Keogh, Brendan. *An Incomplete Game Feel Reader.* Retrieved from: https://brkeogh.com/2017/03/31/an-incomplete-game-feel-reader/

Kremers, Rudolf. *Level Design: Concept, Theory, and Practice.* Wellesley, MA: A.K. Peters. 2009. P. 33.

Maxis. (2013). *Sim City*. Electronic Arts. Retrieved from https://www.ea.com/games/simcity

McREL International. 2014. "Common Core Mathematics 6-8." Content Knowledge: A Compendium of Standards and Benchmarks for K-12 Education: Online Edition. Retrieved from: http://www2.mcrel.org/compendium/standardDetails.asp?subjectID=34&standardID=23.

Minnery, J., & Searle, G. (2014). Toying with the City? Using the Computer Game SimCityTM4 in Planning Education. *Planning Practice & Research*, *29*(1), 41–55. https://doi.org/10.1080/02697459.2013.829335

Mojang. (2011). *Minecraft*. Mojang. Retrieved from https://iminecraft.net

Nebel, S., Schneider, S., & Rey, G. D. (2016). Mining Learning and Crafting Scientific Experiments: A Literature Review on the Use of Minecraft in Education and Research. *Journal of Educational Technology & Society*, *19*(2), 355–366.

Nutt, Christian. (2012, April 13). *The Structure of Fun: Learning from* Super Mario 3D World's *Director.* Retrieved from https://www.gamasutra.com/view/feature/168460/the_structure_of_fun_learning_.php.

Rogers, Scott. *Level Up! The Guide to Great Video Game Design.* San Francisco: Wiley. Second edition. 2014.

Rogers, Scott. "Hell, Hyboria, and Disneyland: The Origins and Inspirations of Themed Video Game Level Design." 2016. In Totten, Christopher W. *Level Design: Processes and Experiences* (101-120). Boca Raton, FL: CRC Press.

Rowe, J. P., Shores, L. R., Mott, B. W., & Lester, J. C. (2011). Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments. *International Journal of Artificial Intelligence in Education*, *21*(1–2), 115–133. https://doi.org/10.3233/JAI-2011-019

Salen, Katie & Zimmerman, Eric. *Rules of Play: Game Design Fundamentals.* Cambridge, MA: MIT Press. 2004. pgs. 336, 351.

Squire, K. (2006). From Content to Context: Videogames as Designed Experience. *Educational Researcher*, *35*(8), 19–29.

Squire, Kurt. *Replaying History: Learning World History through playing Civilization III.* Dissertation. Indiana University. 2004.

Swink, S. (2008). *Game Feel: A Game Designer's Guide to Virtual Sensation*. CRC Press. Retrieved from https://www.crcpress.com/Game-Feel-A-Game-Designers-Guide-to-Virtual-Sensation/Swink/p/book/9780123743282

The Creative Assembly. (2004). *Rome: Total War*. Activision. Retrieved from https://www.totalwar.com/eras#rome_total_war

Totten, Christopher W. *An Architectural Approach to Level Design*. Boca Raton, FL: CRC Press. 2014. pgs. 119-120, 174.

Wardrip-Fruin, N. (2009). *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. (M. Fuller, L. Manovich, & N. Wardrip-Fruin, Eds.) (First Edit). Cambridge, Massachusetts: The MIT Press.

White, Matthew M. *Learn to Play: Designing Tutorials for Video Games.* Boca Raton, FL: CRC Press, 2014.