

ARCHIVES

Cooperative Learning in Educational Gaming

by

Bradley R. Gaffney

B.S., Massachusetts Institute of Technology (2011)

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

February, 2013

© 2013 Massachusetts Institute of Technology. All rights reserved.

Author: _____
Department of Electrical Engineering and Computer Science
January 14, 2013

Certified by: _____
Prof. Eric Klopfer, STEP Director, Thesis Supervisor
January 14, 2013

Accepted by: _____
Prof. Dennis M. Freeman, Chairman, Masters of Engineering Thesis
Committee

Cooperative Learning in Educational Gaming

by

Bradley R. Gaffney

Submitted to the Department of Electrical Engineering and Computer Science
on January 14, 2013, in partial fulfillment of the
requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis analyzes Herbo Hunt, an educational game that aims to teach variables to students. By assigning unknown point values to creatures that must be caught in groups, we subtly encourage students to solve mental equations to achieve their goals. In addition to the formal study, the creation of the game itself utilized intelligent and flexible design techniques. Two main focuses were modularity of design and opportunity for procedural generation of content. Two versions of the game were used to explore the difference in learning between them; we created both a single-player experience and a cooperative version. We expected that players of the cooperative game would learn more quickly through discussion and coordination. Students played both versions, and then answered short questions regarding their experience. Contrary to our expectations students did not appear to show a difference in learning between the two versions; understanding seemed to take place at the same point regardless of game type.

Thesis Supervisor: Prof. Eric Klopfer, STEP Director

Acknowledgments

I want to begin by thanking my family and friends, without whose love and support none of this would be possible.

Thanks to my project team: supervisor Eric Klopfer for helping guide all aspects of my thesis to their final implementations; project manager Louisa Rosenheck for helping with everything from creative direction to logistics for testing; artist Daisy Yuen for bringing the game to life with lively characters and a fantastic feel to all the art; and Jason Haas for working as my initial contact to STEP and bringing me back for my MEng. Thanks also to Chevy Ray Johnston, creator of the Flashpunk game engine on top of which I built Herbo Hunt.

I also wish to thank the MIT Game Lab and STEP for creating and funding this project, and MIT for offering me both a place to further my studies and the provided aid for that education.

Contents

1	Introduction	9
1.1	Problem	9
2	Background and Related Work	12
2.1	Scheller Teacher Education Program	12
2.2	Woosh and Waker	13
2.3	The Radix Endeavor	15
2.4	Animal Catching and Herbo Hunt	15
2.5	Use Of Computer/Video Games for Learning	17
2.6	Learning in MMOGs	17
3	Game Description	19
3.1	Gameplay Overview	19
3.2	Code Description	21
3.3	Early Decisions	22
3.4	Iteration and Feedback	23
3.4.1	STEP Lab Demonstration	23
3.4.2	4/9/12 Playtest	24
3.4.3	4/28/12 Playtest	25
3.5	Evolving Decisions	26
3.5.1	Turn In Mechanic	26

3.5.2	Narrative	28
3.5.3	Multiplayer	28
3.5.4	Procedural Generation and Level Progression	29
3.6	Code Modularity	30
3.6.1	Modularity In Overlapping Sections	31
3.6.2	Screen Sharing - The Greatest Obstacle	31
4	Study Setup	34
4.1	Game Modifications	34
4.2	Study Details	35
5	Game and Study Results	37
5.1	Game Review	37
5.2	Study Findings	39
5.2.1	Likes	39
5.2.2	Dislikes	40
5.2.3	Contrasting the Modes and Response to Variables	41
5.2.4	Conversation	42
5.2.5	Strategies	43
6	Conclusions and Future Work	45
6.1	Conclusions	45
6.2	Future Work	46
A	Herbo Hunt Level Progression	50
B	Demonstration and Playtest Feedback	52
B.1	Demonstration Notes	52
B.2	4/9/12 Playtest Feedback	53
B.2.1	Playability	53

B.2.2	Controls	54
B.2.3	Information	54
B.2.4	Design	55
B.2.5	Style	56
B.3	4/28/12 Playtest Feedback	56
B.3.1	Observer 1 Notes	56
B.3.2	Observer 2 Notes	57
B.3.3	Discussion Responses	58

List of Figures

3-1	A GameMap, populated with various entities	19
3-2	Players at opposite corners of the map	32

List of Tables

A.1 Levels	51
------------------	----

Chapter 1

Introduction

On the surface, Herbo Hunt is a simple game of catching colorful creatures. Under that shiny exterior, students are learning about variables and solving equations without even noticing. This thesis analyzes two variants of the game: single player and cooperative. It will look into both the creation and development of the games, as well as a formal study that attempts to determine whether one version facilitates better learning of the material.

1.1 Problem

Educational gaming is a concept that has so much promise. Since these games have muscled in on a large part of the entertainment market [4], it would be convenient for educators to make use of these games to teach to their students. Early attempts have not been as well received as those in the education sector would have hoped [6]. Most educational games today have a different atmosphere about them. Players often do not immerse as well as they can in other games, and the learning is surface level. With the teaching out in the open, some of the students that teachers hope to reach with this medium are turned off by it [3].

On top of that, educators and game developers could work more efficiently as a

team if they knew which aspects of educational gaming allowed for better learning. It is not only a matter of teamwork; the feasibility of continuing research in games depends on finding out how to implement them effectively. Our team wanted to experiment with stripping the research down to a single variable. We wanted to see whether players could learn concepts more quickly in a cooperative multiplayer game or in a single player version. In developing Herbo Hunt, our team saw the opportunity to create both a single player and cooperative experience that would not require drastic differences between the two versions.

In order to produce these two versions in the most efficient manner, we imposed a few design goals on the game itself. Most importantly, the game needed to be designed with both versions in mind. All design decisions needed to continue to allow both a single player and a team of two players to enjoy the game in its full capacity. Not only did the design need to let both games coexist, but both games needed to operate as similarly as possible in order to truly see whether the multiplayer option affected learning. In other words, the games overall needed to be as much of a control as possible, with the exception of using the differing number of players. Not only would this allow for better scientific control, keeping the two versions would allow for modifications of the game to occur much more smoothly, because most factors in the game would only be located in a single place rather than split across two different sections.

The next section will discuss the background material that contributed to the work of this thesis. After that, the game will be described at a high level as well as provide some explanation of the code structure. The game description will also cover the goals of the game's creation and some of the features that evolved as the thesis progressed. Next the setup of the study will be described, including modifications to the game that were made specifically for the study portion of the thesis. The results of the study will be given, which will be composed both of a critique of the final game

and the responses of the students to various questions. Conclusions will be provided after that, and finally extensions will be discussed.

Chapter 2

Background and Related Work

2.1 Scheller Teacher Education Program

The Scheller Teacher Education Program (STEP) has been teaching MIT students to teach since 1995. Most MIT students have a very deep understanding of math and science. However, not many of them intend to teach others how to develop that same understanding. STEP believes the select few students that are interested in spreading their gift to others can become some of the most powerful and helpful teachers. Not only do MIT students have a strong basis in math and science, they have also learned to test and break their own limits. Using this mentality, teachers that have MIT degrees are able to push their students past obstacles that seemed unsurmountable [5].

In addition to teaching MIT students to teach, STEP has taken part in the production of many educational games and simulations. One of its more well-known applications is the StarLogo development environment. Inside StarLogo, users are able to add objects into a world, and code behaviors for them using coding blocks provided by the StarLogo interface. These blocks allow users to not worry so much about syntax checking, and they can instead really experiment with different ideas in

programming.

Although StarLogo is very popular, it is clear to the user that they are performing educational experimentation. Another endeavor that STEP often undertakes is the creation of educational games. These games tend to have a similar approach to learning. The games are meant to be a truly fun and immersive experience, and seamlessly weave the teaching into the gameplay. The desire of many of these games is to have students play and enjoy the game first, and then when prompted to reflect suddenly realize that they have learned a new concept. Members of the STEP lab believe that this approach helps the students remove the mental connection between their learning and playing of the game. Dissolving this perception allows the student to develop deeper understanding, because they are more willing to delve more deeply into the game itself and experience more of the teachings.

2.2 Woosh and Waker

Beginning in 2009, the MIT Game Lab collaborated with members of STEP and the National University of Singapore to create Woosh and Waker. Woosh and Waker were two varieties of the same game. The games taught physics concepts to its players while trying to avoid explicitly teaching students, as is the case with many games produced by STEP. In this particular game, the player moved from map to map by reaching the end portal. Each level had one or more orbs and one or more graph regions. Once the player picked up the orb, a line would begin moving from left to right on the graph. The player could move to influence the graph as it traveled. There were two variants of the graphs: displacement vs time and velocity vs time. For example, in the case of a displacement vs time graph the player could create a slope by picking up the orb and running to the right for the entire duration of the graph. Initially, all of the graphs were displacement graphs. As the player progressed and mastered that

concept, he or she moved on to velocity graphs.

The focus of the study was not to simply create a new educational game for students to play. The creators made four different versions of the game. The first was Waker 1.0. Waker 1.0 had story at the beginning and end of the game and between levels. There were two versions of Waker 2.0; the first simply had an extended narrative to Waker 1.0 while the other had an extended narrative, but also allowed players to replay dialogue they heard. Finally, Woosh was an abstract version of Waker. It had absolutely no story to it, and even the character and objects were redesigned to prevent storytelling on the part of the game.

The study gathered students attending High School in Singapore. It separated the students into 4 groups, each group playing one version of the game. After playing the game, the students answered questionnaires rating various aspects of the game on a 1-10 scale. Students found the Woosh version to be almost universally the worst of the versions. In most cases increasing the amount of narrative and opportunity for repetition provided a more enjoyable experience, although there were some categories in which the extended narrative scored better without the addition of the ability to review what had been said.

In addition to the findings regarding levels of narrative, the research teams found a difference in experience resulting from the inclusion of an off-screen child character that would narrate and explain things to the player. The researchers found that players had an easy time connecting with the off-screen child and become more involved with the story. They also found that by having a child's voice interacting with a child player, the students saw the voice as a peer and more easily took its advice and, as a result, learned from it.

This study did not just produce interesting results regarding the use of narrative. The differing versions had as much in common with each other as possible except for the tested element. This approach gave STEP a new method of exploring the

usefulness of various aspect of gaming with respect to teaching educational concepts. Using small games with limited variation, STEP could make much of the versions constant and only differ in the aspect that a particular investigation wanted to explore.

2.3 The Radix Endeavor

In the spring of 2011, STEP began exploration of a new concept in educational gaming. STEP began working on an education massively-multiplayer online role-playing game (MMORPG, or MMO). STEP wanted to apply the same principles of learning paired with a deeply engaging and interesting game, while attempting to take advantage of the community of an MMO. MMOs are well-known for their society, bringing together people from all around the world to compete and collaborate in a digital setting. STEP wished to utilize this environment to foster group understanding of educational concepts, particularly those in science, technology, engineering, and mathematics (STEM). Originally titled STEMMO, this project has since become known as the Radix Endeavor. The Radix Endeavor has a target audience of high school students, given the highly social nature of MMOs and the wide range of subjects covered in a high school curriculum.

2.4 Animal Catching and Herbo Hunt

As a design team worked on very early design stages of STEMMO, they spent a considerable portion of time working on the introductory world. This area would refresh players on the concepts that they had learned throughout middle school in preparation for high school. All four of the branches of STEM have a basis in mathematics, and therefore the starting point for this introductory world was to create quests to explore with variables.

One concept, tentatively named ‘Animal Catching,’ had players catching animals

on a small map. The players had been asked to enter the yearly tournament by the king of the current land. The king asked the player to enter because the king himself was forbidden from entering. The king would get the player into the competition, but the player then had to work within the constraints set forth by both the competition and the king himself. The player had to achieve various goals for each round by catching animals worth various point values. However, the values for each animal were not explicitly stated. Furthermore, the player would sometimes have to catch more than one animal at a given time in order to turn in his current set. This required the player to set up equations and solve them to meet his goals. Since the math required was not very complicated, the players could set up and solve the equations mentally without breaking the flow of gameplay. This sidequest is what eventually evolved into the full standalone game of Herbo Hunt.

Recently, STEP has been working in collaboration with the MIT Game Lab performing research on approaches to educational gaming. The Woosh/Waker project is one of the products of this research. With the observed success of the approach taken by Woosh and Waker, we wanted to use the same method of small games with little variation. Our team decided to use the Animal Catching idea, and develop a standalone game from it. Originally we had intended to explore narrative with this study just as had been done with Woosh and Waker. However, as our team progressed further with the project we shifted our focus. Part of the reason that the Radix Endeavor is an MMO is to explore the use of a community within an MMO to reinforce educational concepts. Since Animal Catching came from the Radix Endeavor, we considered altering the study to look at multiplayer. As design and implementation progressed, multiplayer worked even better as the focus of the study. It proved to be better for code modularity and allowed the storyline to affect the design of all versions of the game in the same way. With all of these factors, we changed from exploring narrative to testing the effects of single player and multiplayer modes.

2.5 Use Of Computer/Video Games for Learning

Many other publications have examined educational games, the effect of cooperative gaming, and some have even examined the two at the same time. In *The Use of Computer and Video Games for Learning* [2] Mitchell and Savill-Smith have scoured available literature to find a set of reasons for the usefulness of games as a learning tool. The engaging nature of video games pulls players in using immersive environments, and gives them a goal to keep players striving towards something. They found that games allow players to participate in activities that were otherwise impossible or inaccessible. On the other hand, games can present players with very lifelike situations and give them the opportunity to manipulate the scenario.

Mitchell and Savill-Smith cover an extensive amount of previous literature regarding video games and their potential use in education. They discuss papers exploring new avenues and opportunities in educational gaming, such as handheld gaming, improved artificial intelligence, and the use of multi-user environments. They also discuss possible risks mentioned in various literatures ranging from a mismatch in goals to alienation of students from the real world.

The review and exploration performed by the authors of *The Use of Computer and Video Games for Learning* covers a broad range of topics regarding educational gaming and its uses. The review covers many reasons supporting video games in education, but does warn against some potential pitfalls. Some of the article discusses multiplayer gaming, and provides further references to other papers that cover the same topic.

2.6 Learning in MMOGs

In a 2004 article, Constance Steinkuehler [4] discusses how players learn within the community of an online multiplayer game. She shows that videogames are one of the

most profitable industries in the world, and that massively-multiplayer online games (MMOGs) are working into current culture. Steinkuehler explores the communities in MMOs and tries to learn what they might have to offer us as a society.

The article is driven by questions concerning what players must do to be a functioning member of the MMO community, as well as the effect that participation has on the real lives of the players. She analyzes a particular in-game interaction between a pair of players. One player is significantly more experienced than the other, and teaches the less informed player about some of the rules within the gaming society. Without explicitly stating it, the veteran player utilizes many teaching techniques to inform the player about the world and its norms.

Steinkeuhler's early findings have some promise for learning in MMOs. She finds that the full experience is only obtained through interacting with others. Also, the system of learning has a reflection and feedback built in. If a player fails a mission or the character is defeated, recovering and trying again is not a very difficult process. However, she does warn that these contribute to learning about the game; extending these MMOs to teach other concepts requires more attention than simply adding math problems into an existing game.

Chapter 3

Game Description

3.1 Gameplay Overview

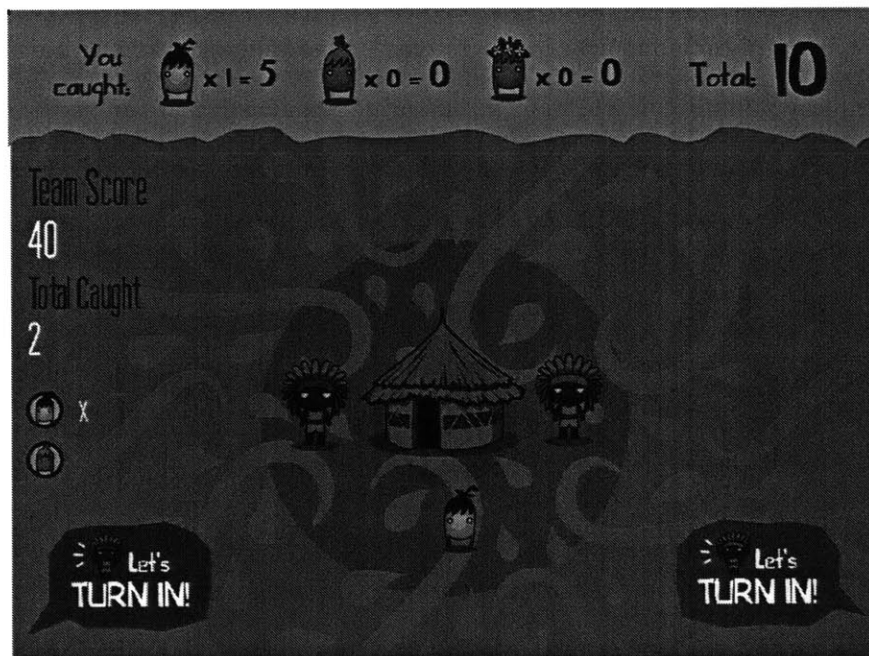


Figure 3-1: A GameMap, populated with various entities

At the start of Herbo Hunt, the player is given a choice of avatar, or avatars in the case of multiplayer. The story displays, and gives the background for playing the

game. The story explains that people of the town have fallen ill. To restore them to health, the player must catch Herbos, the animals in the area. These Herbos can then be exchanged for medi-nuggets that can be given to the sick townspeople.

The player is then immediately thrust into the first level. A quick tutorial flashes up for the player, explaining in pictures what keys perform what actions. The selected avatar is now on a map. The level goal sits at the top of the screen. On the side the player can find his or her score, the number of Herbos turned in so far, and the amount of Herbos the player has currently caught. The actual game map consists of the avatar, various Herbos, and the medimaid hut.

The first thing the player needs to do is catch a Herbo. By moving close to a Herbo and pressing the 'catch/turn in' key, the player removes the Herbo from the map and updates the count of Herbos currently caught. If the player then tries to catch another Herbo, he or she is given a popup message that says "Bag Full!" The Herbo does not disappear from the map, and the player realizes that this first Herbo must be turned in.

To turn in, the player returns to the center of the map, where the medimaid hut sits. Here, the player again presses the 'catch/turn in' key. The screen flashes and the top banner changes to show the player the number of medi-nuggets (in other words, points) that have been earned with this particular turn in. The scores and amount caught on the side banner update, and the player is free to catch another Herbo.

There are some notable changes as the game progresses. Some of the levels change the point values for the Herbos. The number of Herbos that can be caught before turning in increases from 1 to 2, but it is not an optional increase. The player must catch 2 Herbos in order to successfully turn in. At the very end, opponents are introduced into the game and challenge the player to reach his or her goal more quickly or efficiently than the opponents. The full progression of levels can be found in Appendix A.

3.2 Code Description

The Flashpunk engine provides a lot of the functionality to make a game function. There are two high level classes from which almost all other classes inherit: world and entity. Only one world is active at any given time, and a world can have any number of entities within it. Entities do not contain other entities. Flashpunk is a frame-based engine, meaning that every second is divided into a certain number of 'frames.' Each frame every active entity is allowed to act once, performing what is known as its 'tick' function. The world also ticks, and the game itself has some functionality it performs each frame.

Most of the game is spent within a GameMap world. This GameMap adds the Player entities, the Monster (Herbo) entities, the Turnin (Medimaid Hut) entity, and everything else on screen including the background and scoreboard. The GameMap is created from a set of parameters specified by the current series and level. To save some information between GameMap instances, there is a set of global variables that can persist throughout the game, and can be saved off to restore the game later.

There is a lot of interaction between the world and its entities, and little direct interaction between entities. The only situations that involved entities reacting to each other without the intervention of the world were collision detection and some player functions. Flashpunk provided the underlying detection method, but the overarching idea was simple. Every entity could be given a type and a size. To check for a collision the Flashpunk engine would iterate through all entities on screen and see if the bounding box given for the entity in question overlapped with the bounding box for another entity of the specified type. During collision detection a player could also catch a Herbo by pressing the correct key, and could turn in caught Herbos if collision was detected with the medimaid hut.

Aside from collision detection and functions involved with it, all interaction between entities was performed through the world. Even though the player caught

Herbos without the help of the GameMap, the scoreboard did not update. This allowed the player to have no concept of the scoreboard. Instead, the GameMap was constantly checking the player's believed score, and sending that score off to the scoreboard entity. In the case of two players the GameMap would check both scores, add them, and send that score off to the scoreboard.

3.3 Early Decisions

Most of the only decisions that stayed solidified from the beginning revolved around general gameplay mechanics. Some of them resulted from the limitations of the project. The gameplay would take place on a two-dimensional map with a top-down view. This allowed the player a lot of freedom to decide where to go next to catch a Herbo, but did not require the power of a 3D engine. Animations were restricted to simply those of player, opponents, and Herbos moving around the map. Animating more features of the game would have required more time from our artist than acceptable.

Other constant design decisions resulted from the setup of the game itself, and its need to convey information. The screen layout would include information bars that would tell the player as much as the team deemed relevant. Although the information on the bars would change as the project evolved, the need for the bars never dissipated. Another obvious constant was to have the Herbos to catch, the player to move, and opponents to play against. Finally, the concept of each individual game being contained on a single map with a single particular goal did not change as the project evolved. Although the games became grouped into series, individual games all had the ability to stand alone.

3.4 Iteration and Feedback

During the creation and evolution of Herbo Hunt, our team had three separate sessions where we gathered feedback for improving the game. Our first session was a demonstration for other members of the STEP lab. Since all of them had some background in educational gaming and this was early in the game's production, we received a lot of useful suggestions regarding the direction of the game and ways to improve its mechanics. Later on we had two playtests with middle school students. We used these sessions to further fine-tune the game itself, but these sessions also provided feedback regarding the effectiveness of various aspects of the game. Our fully detailed notes can be seen in Appendix B.

3.4.1 STEP Lab Demonstration

On February 1st, 2012 we showed Herbo Hunt to the STEP Lab. We gave a short overview of the game's intentions and demonstrated the game in action. The idea was well received and the testing approach sounded feasible to the members of the lab. The subsequent conversation covered many different topics, but some ideas provided very useful insight and some were incorporated into the final game.

One big focus of the discussion was to ensure that the learning progressed in a way that would best benefit the players. Many of the suggestions regarded "scaffolding," in which changes to the game mechanics were introduced individually. Also, when the game introduces a new mechanic it can reset other components of the game so that the player can more easily understand just the new concept. For example, when the bag size increased from 1 Herbo to 2, we could reset the number of different Herbos back to just one. This approach would allow us to ease the player into every change in the mechanics, but it would also add length to the overall game. We focused on the slow introduction of components that scaffolding provides, but we did not reset

everything on changes.

The topic most heavily discussed regarded the turning in of Herbos. In the demonstration the player had no control over when Herbos turned in. Herbos were caught and immediately updated the player's score, but the scoreboard did not refresh until a particular time interval had passed. Many members of the lab felt that this took away some of the control from the player. It was suggested to have a 'basket,' and the player had to choose to turn in the basket when it filled up. The basket seemed very beneficial and well supported, so our team incorporated that concept into the further and final iterations of Herbo Hunt.

3.4.2 4/9/12 Playtest

We performed our first playtest at an after school program for middle school students. We brought students into our testing area two at a time. At this point our focus was not to gather data for the experiment; we simply sought feedback on the game in general. Because of this, players only played the single player version of the game. After players finished, we asked them a few questions to elicit more feedback than could be obtained by just observing.

The most apparent issue we discovered through this playtest was the fact that the students were not given enough information to fully understand the game. Although we did not want to weigh students down with a lengthy tutorial and we did not want it to be obvious that this was an educational game, we erred too far on the withholding side. Players had many questions regarding what key to press or why they had failed a level. From this we saw that we needed to make the goal more explicit, make catching and turning in work more smoothly, change the controls to be more intuitive, and provide players with more ways to recall information.

Players also provided a number of insights regarding the progression of the game itself. Students wanted a stronger sense of progress, suggesting that the setting change

as levels are cleared, or allowing the player to use collected points to purchase upgrades for the avatar. The players enjoyed the changes to gameplay mechanics, such as the map size and bag capacity increases. The students liked the avatar and monsters, but some did suggest that the characters look happier and more pleasant.

3.4.3 4/28/12 Playtest

Our second playtest took place on the campus of MIT. We invited a number of middle school students in the surrounding area to come to MIT and play the game. The students all played simultaneously, and two members of our team took extensive notes. Players began by playing individually, and then paired up to try out the multiplayer version of the game as well. After all of the students had played for their allotted time period, we led a discussion with them to find other possible improvements to make before our study.

Since this playtest was later in the game's development, we were watching to see if this game was meeting our original goals. One particular student said that she did not enjoy mathy games, but this game didn't seem mathy. Responses like this gave us confidence that Herbo Hunt was providing learning without feeling too direct. Players also liked the core mechanic of catching Herbos for points, and playing with the various level goals.

There were also some gameplay features that we saw needed modification. When we performed this playtest players received a popup message every time they turned in. This message told players how many Herbos of each type they caught and how many points they earned for a particular turn in. This message had to be dismissed every time before play could resume. Players found this useful at first, but obtrusive as the game progressed. They mentioned in the discussion that they felt slowed down by the popup. They especially disliked this when playing together, because one player turning in stopped gameplay for both players. We feared this slowdown would affect

the learning process, so we modified the turn in to appear at the top of the screen and not block progress.

The multiplayer version in particular received some complaints. One pair of students said that they found the keyboard to be too crowded. Unfortunately, we did not see any remedy for this early on. Herbo Hunt did not have online functionality so our multiplayer needed to be local. During the study we realized that a second USB keyboard could be used to play with some more room, and incorporated that since we did not believe it to be a factor of our study. A different pair of students was bothered by the crowding of the characters onscreen. The players would collide with each other, and they could not leave the shared screen space. However, the Flashpunk engine had issues with allowing players to pass through each other, so we could not change that aspect of the game. We also did not wish to change the shared screen space concern due to the fact that we needed to keep both players on some screen and implementation of a splitscreen mode was not feasible.

3.5 Evolving Decisions

As this project took 2 years between inception and completion, many factors of the game evolved as the goals and needs of the game became more well-defined. Some aspects changed their priority within the overall structure of the game. Others were modified in their overall operation to better fit the different iterations of the game. Some became modified as a result of changes to the plans for the study itself.

3.5.1 Turn In Mechanic

Initially, the player had no control over when to turn in Herbos. The Herbos only turned in when the target number was caught, or when a particular amount of time had passed. The target number was not the goal score or number of Herbos, it was

instead the 'bag size' for the player. This was the mechanic that made players think more closely about the point values, and use their equation solving skills to determine the point values. In the case of the timed refresh, players had to decide whether to catch a few and wait or catch many and take their chances.

After gathering suggestions from other members of the STEP lab, the turn in mechanic became less automatic. Players had to return to a central location and press a key to turn in their Herbos. They could only turn in when they had reached at least the minimum, and once they had caught their maximum they could not catch any more until they turned in. Additionally, we decided to implement drop functionality so that the player could release the Herbos they had caught without getting their points.

Even after we had decided that turning in should be manually performed, the turn in procedure went through a number of iterations. Initially, the player simply had to press the designated 'turn in' button. To increase the challenge, we required the player to turn the Herbos in at a specified location on the map. Initially the player turned in at a simple stamp that had 'Turn In!' written on it. Although this stamp made it very clear to the player where he or she needed to turn in for points, the image did not fit well with the style of the game. In order to make the turn in location more appropriate for the game's style, we changed the image to a satchel. This satchel matched the appearance of the other game objects, but it did not make sense with the game's story to put the Herbos in a bag in the middle of the playing field. Once we developed the story we had a clear reason for turning in. To match the story, the satchel became a medimaid hut where the player would perform the exchange of Herbos for medi-nuggets.

3.5.2 Narrative

The narrative began as the focus of this experiment, much like with Woosh and Waker. As our team worked on the game, however, the exploration of multiplayer vs single player seemed to be a more beneficial study. This decision came with the realization that Woosh and Waker had already tested narrative, and Herbo Hunt could very easily be no different in its experimental procedure. In addition, creating the narrative for Herbo Hunt was not as programming-intensive as other aspects of the game. The narrative required extensive thought, but not much work from an engineering perspective. After the fall semester of 2011, our artist Daisy Yuen picked up most of the design responsibility. As a result, the narrative took a backseat, and ended up being the exact same across the single player and multiplayer experiences.

3.5.3 Multiplayer

As previously stated, multiplayer was not the focus of the study at first. Also, the multiplayer initially implemented was competitive rather than cooperative. As game development progressed it became clear that allowing one player to succeed while the other failed was nonsense on a single computer, since the players would move on together or not at all. In addition, some of the scoring mechanics developed for the single player progression did not make sense in the competitive variation of the game.

Not only did the competitive multiplayer not fit with the scoring mechanics, competitive multiplayer games create a different dynamic between players than cooperative games. The competitive nature could easily cause problems in a classroom environment, which would quickly remove Herbo Hunt as a viable means to teach variables. Given all of these concerns, we decided to make the multiplayer cooperative. This decision was actually the first step toward studying the learning differences in single or multiplayer versions.

Although competitive multiplayer had many drawbacks, our team saw promise in

cooperative multiplayer. We thought that a cooperative experience could encourage more communication between players. Through this communication players could discuss strategies for overcoming the obstacles set up against them. This interaction was not limited to just during the rounds; players could also reflect after the level through discussion with each other. Evaluation of strategies could be performed collaboratively after a round, letting players discover new ideas more quickly. Cooperative multiplayer also allowed for better modularization of the game code. By making the multiplayer cooperative, the level goals could stay consistent through both modes. This eliminated another variable in the testing procedure, allowing the testing to more closely follow the model set forth by Woosh/Waker.

3.5.4 Procedural Generation and Level Progression

Procedural content generation is a method that creates some part of a game randomly using a set of specified parameters. An example of this can be found in Blizzard's Diablo. Dungeons in Diablo have number of levels across all plays, but the map layout and the goals for a given floor change from play to play. Procedural content generation is a popular aspect of many games, as it allows users to avoid playing the exact same game every time [1].

We originally wanted Herbo Hunt to produce its levels procedurally, with no set progression. However, as we explored our options with the game, we found that a set progression would better control the external factors given that we only wanted to test the difference between single and multiplayer versions. In addition to setting a progression overall, we wanted the ability to have levels in series. These series would have the same point values for Herbos, so a player could learn the values in one level of a series and utilize that knowledge in subsequent levels. Often the first level of a series would have a simple, possibly even impossible to fail, goal. This level was intended as a practice round for the player to just figure out what the point values

had become. For example, the first level of the second series had two Herbo types worth 10 and 5 points. The player needed to score more than 45 points, so there was no real restriction on which Herbos could be caught and the player could not fail by catching the wrong Herbo. On the second level of the second world the player needed to reach exactly 100 points. If the player had used the first level to figure out point values, he or she could perform the math to determine how many of each Herbo to catch. We strove for this approach in each of the worlds.

3.6 Code Modularity

Code modularity is a term that means to have different pieces of code that could work independently of each other. This way, programmers can mix and match the pieces of code used to obtain desired results. Code modularity paves the way for robust code creation, as it separates each piece from the others as much as possible. With this strong separation, different parts of the code affect each other as little as possible, and as a result issues are often centralized to a single piece. This leads to shorter debugging times, as the problem does not need to be traced all through the code base.

Particularly in Herbo Hunt, we strived for modular design. Most of the different objects in Herbo Hunt are contained within their own class, and are all brought together by the GameMap class. Aside from the overarching control of GameMap, the different objects in Herbo Hunt have little to no interaction. As previously mentioned, modularity made debugging easier, but it also simplified the creation of the two game modes. Very few classes within the code need to be aware that there is a second player; almost all of that is handled within the GameMap. Although there was a necessary overlap between the classes, by minimizing it we made the coding process run more smoothly.

3.6.1 Modularity In Overlapping Sections

The Player class needed to be aware of the possibility of a second player. Without that awareness, there could be no distinction of control from character to character. A variable of playerNum was assigned along with the creation of player. This value was checked along with keypresses to determine whether to have the particular player react to keys. Aside from that, the playerNum variable was never checked within Player. As with the other classes, Player left it up to the GameMap to deal with most of the single vs multiplayer distinctions.

GameMap controlled nearly all of the distinctions between the single and multiplayer versions. GameMap populated the screen with all players, opponents, and Herbos. The scoreboard was another distinct entity on the GameMap. The scoreboard also had no sense of multiplayer, GameMap did all of the math to make the players work cooperatively. Due to the cooperative nature of the gameplay, GameMap did not have to do much work to make the scores combine. In most cases all that was necessary was to simply add the scores for player 1 and player 2 together, and submit them as one score. GameMap also passed the scores and number of Herbos caught to the win/loss evaluation function so that all it needed to do was evaluate as if the game was single player.

3.6.2 Screen Sharing - The Greatest Obstacle

In order to provide code functionality that easily translated between single player and multiplayer, we began by defining variables CameraCheckX and CameraCheckY. The camera coordinates of the Flashpunk engine specified where the top-left corner of the camera should be located. The single player camera was simple to define. We wanted the camera to stay centered on the player, but not go outside the boundaries of the map of the game. CameraCheckX and CameraCheckY Therefore, in the case of the single player code CameraCheckX and CameraCheckY were simply equal to

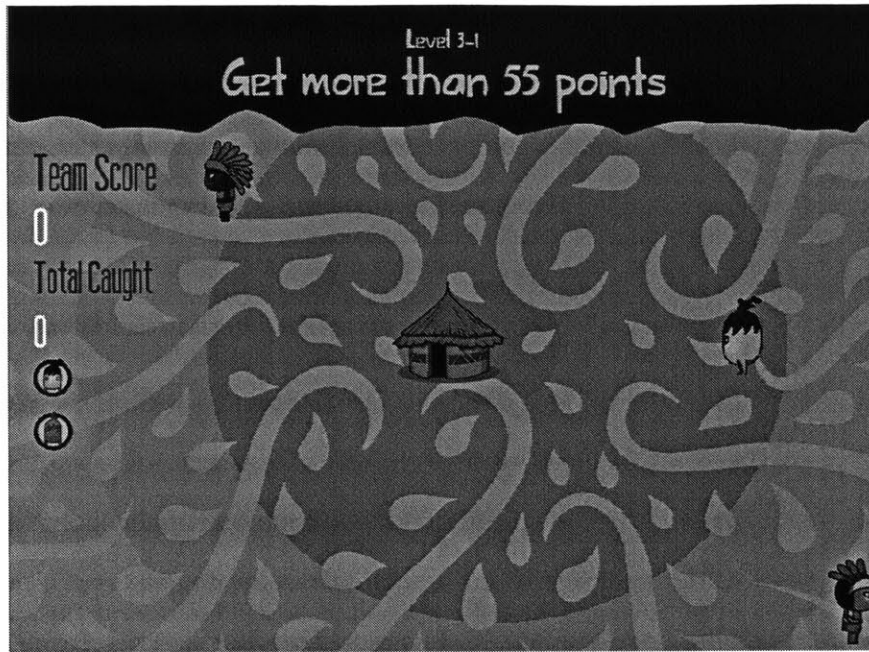


Figure 3-2: Players at opposite corners of the map

the X and Y coordinates of the player.

In the case of the two player version, CameraCheckX and CameraCheckY proved much more difficult to calculate. Rather than pick one of the two players and make him or her the center, CameraCheckX took the midpoint between the two players. However, this introduced the need for a very careful balance. The players needed to be able to move the midpoint, but at the same time they needed to be restricted if either of them tried to move offscreen when the other player was on the other edge. Like the camera, the x and y coordinates of the players corresponded to the upper left corner. In order to obtain the midpoint correctly, we used the following code:

```
if (player1.x > player2.x)
    cameraCheckX = ((player1.x +
    player1.getDimension(0) + player2.x) / 2);
else
    cameraCheckX = ((player1.x +
```

```
player2.getDimension(0) + player2.x) / 2);
```

CameraCheckY was calculated in similar fashion. `getDimension(0)` returned the width of the player object. To get the true midpoint, we first needed to find the center of the player objects. Then, we could take the average of those two x values.

However, even once we had calculated the correct CameraChecks, we still had to modify the value so that the camera would line up correctly. We had to first move the camera to the left by half of the width of the playable game screen, as well as up by half the height of the playable game screen. However, the camera couldn't move outside of the game map. This resulted in the need to make sure x was greater than or equal to 0 and that x was less than or equal to the size of the map minus the size of one game screen. Therefore, finally calculating the x coordinate of the camera required this manipulation of CameraCheckX:

```
FP.camera.x = Math.max(Math.min(  
    cameraCheckX - Globals.GAME_SCREEN_SIZE_X / 2 -  
    Globals.SCOREBOARD_WIDTH, mapSizeX -  
    Globals.GAME_SCREEN_SIZE_X), 0);
```

Despite how simple the concept of the shared camera seemed, it actually proved to be one of the most troublesome aspects of gameplay, especially on code changes. Since CameraCheck was so strongly dependent on many factors and there was little leeway for incorrect camera movement, it was very easy to break the camera system with even a single one-off error. Typically this would result in one player having the ability to drag the other player when they were on opposite ends of the screen.

Chapter 4

Study Setup

4.1 Game Modifications

In order to make the game more isolated for a controlled experiment, we made some modifications. The first change regarded character profiles. In the original formation of the game, we allowed for saving and loading of content. Players would enter a username when they began playing, and the game would automatically save as they progressed through. The player could then come back to the game later and resume from the point where they stopped. However, the experience we wanted the students in the study to have would be more of a single session. As a result, we decided to eliminate the character profiles for the sake of the study. This had two benefits: first, it eliminated all confusion to the player about choosing and remembering a username; second, it removed the character profile screens from the game and gave players the opportunity to get into the game more quickly.

In addition to removing user profiles, our team also decided to modify the game selection process. Initially after starting up the game users would select whether to play single player or multiplayer. Since this was the focus of the study, we decided to take control from the player regarding number of players. Before the experiment, we

separated the two versions and created fully separate executables. Thanks to modular design, the only difference in the code base was a single variable change from 1 to 2. We also cut down the level progression so that we could follow the players' progress more precisely. Shrinking the progression also prevented the players from accessing some incomplete levels.

4.2 Study Details

The study was carried out over the course of three days of study. We invited middle school students from the area surrounding MIT to participate in roughly a half hour of study. Throughout the study we had a total of 20 students. Some of the students participated alone, while others came to the study in pairs. In the case of pairs, the students seemed at least familiar with each other and in most cases appeared to be friends.

In roughly half of the sessions the players began with the single player version. If there were two students, they each played the single player version simultaneously on different machines. After about 15 minutes of playing the single player version, the players would come together to play the cooperative version. In the case of single students, they played with project manager Louisa Rosenheck. The other sessions proceeded in reverse order; players played cooperatively first, and then separated to experience the individual version. Louisa did not play the single player version, we felt that it would not add to or affect the experiment. After completing their time with both versions of the game, we asked the students a short series of questions about aspects of the game.

We began by asking, very generally, about what the students enjoyed and what they disliked about the game. This served as both an informative question to the team for extension possibilities as well as giving the students a less formal question

that could be seen as a warm-up question. After that we asked students which version of the game they preferred, and why. This question could be answered in any way the student deemed appropriate. Some of the students even preferred opposite versions of the game for the same reason. We then asked more specifically which version was easier to win. After getting information about preferences between the games, we asked students what strategy, if any, they applied to playing the game. We used this question to see how actively the students had been thinking about variables and equations. Finally, we asked directly whether they felt that the game related to topics the students had covered in school. With this question we hoped to bring students to the realization of just how much math they had been using.

Chapter 5

Game and Study Results

5.1 Game Review

This will begin by providing thoughts on the finished product of Herbo Hunt as of the conclusion of the study. First, there will be a discussion of the aspects of the game that could use improvement. Without the procedural level generation the game only has 7 short worlds to experience, and then nothing follows. Even if the procedural generation had been included, the pool of options that can be shuffled is not very large. This results in largely the same level repeating with too slight of a modification to be noticeable. As the game currently stands, there is a single path. This game corrals players in one particular direction and gives them no choice regarding their progression.

Overall, Herbo Hunt is a small game. The narrative does not continue throughout the game with the same force with which it begins. All of the story exists in the introduction. The game has no development between levels, and no conclusion to the short scripted progression. There is no variation in level design other than map size and color. There are very few Herbo types, and opponents were not given an art pass that the rest of the game elements received. After a single playthrough, the player

has experienced the entire game. There is nothing different if a player plays the game a second time.

Despite the issues with Herbo Hunt, it has many good qualities as well. Nearly all the students found the game to be fun and engaging. The students played for their full sessions and did not tire within the half hour. Although 30 minutes is not much time, there are games on the market that would not be able to hold the attention of gamers for even that long. Even after completing the study portion with students we had a number of them return to the computers and continue playing. On some occasions students attempted to continue playing during the discussion portion of the study.

Another very high point regarding the final product that is Herbo Hunt is the final look. The art of Herbo Hunt has been so well done, and unifies the game under a single theme. The players connect with the onscreen characters, and the Herbos are well-designed to be distinct while still fitting the mold. Not only has the overall art brought so much life into the game, there are underlying design choices that made improved the game in ways that most players wouldn't even notice. The layout of all of the screens has been optimized to be both aesthetically pleasing and efficient in its delivery of information. The best example is with the floor of the game screen. The leaves on the ground look like a simple design, but actually all lead to the center of the screen, where the turn in hut happens to be located. None of the players mentioned noticing this, but as an observer we saw that it could be of use.

The design and resulting code for the game stayed very modular throughout the game's creation, as desired. The code is very separable, as most elements can be used individually. Conversion from one mode to another only required the change of a single variable. That one variable only caused code duplication in a small number of locations within the code, and it was only when necessary. Places such as score calculation and player navigation are some of the only examples where the number

of players affected code functionality. Almost every different type of object within Herbo Hunt has its own class, making it easier to track down and fix issues without concern that fixing a problem for one object will break operation of another.

By the conclusion of the game's development, the procedural generation goal had been pushed aside to allow for a more rigid game progression that would structure the study. Random game generation is still possible, but not currently in the game and not finely tuned enough to create an enjoyable experience each time. The biggest drawback to this aspect of the game is the fact that the game has a linear progression with no motivation to replay once a player completes the short set of levels provided. Unfortunately, even with random generation the set of parameters that are used for generation do not result in a widely varied set of games, so procedurally generated levels may still not even add much lifespan to the game without bolstering the game using more options. Such enhancements will be discussed in a later section.

5.2 Study Findings

5.2.1 Likes

The responses of the students had large variation when it came to the more open-ended questions, but was fairly standard on the questions that were more direct. Students enjoyed features of the game that resulted from the game's design. The art received the most praise in general from students, with nearly 1/3 of the students noting it as something they liked. We observed that players took avatar selection seriously, indicating that it was a feature of the game that players cared about. Two of the students mentioned the avatar selection in particular as something they liked. Finally the students found the game to be easy to play, and liked the base mechanic of catching the Herbos and returning them to gain points, especially since the avatars were fast enough to catch up to the Herbos. Half of the students stated that they

enjoyed the underlying game mechanic of chasing and catching.

Regarding overall gameplay, students stated that they found the game to be engaging. Out of 13 recorded responses, 7 sessions had a student or pair of students that liked the math aspect of the game. They were pleased by the challenge offered by the game, and although there was difficulty it was not impossible. Students responded positively to the use of math and the fact that the game required a bit more concentration than games that seemed similar. They attributed this focus to the constantly changing level goals and Herbo point values, and were happy for both of these things. The subtraction Herbos in particular were a highlight for one student.

5.2.2 Dislikes

Some students were also provided constructive criticism regarding the game. Most responses fell into three categories: game programming issues, problems with game mechanics, and issues with aspects resulting from design decisions. Programming issues stemmed from implementation choices and limitations of the Flashpunk engine. All objects in Flashpunk require a rectangular border for collision detection. As a result, players could get stuck on edges where it seemed that nothing was colliding. 3 of the 13 sessions had students mention this as an issue. Also, the Herbos picked for a given level were randomly chosen from a particular subset on each level. However, this did leave the possibility that not enough of the desired Herbo would be placed in the level. Sometimes an entire Herbo type did not make it on to the map, requiring a restart. This came up for 2 of the 13 sessions, and both disliked it.

Some of the complaints regarded mechanics of the game that were explicitly decided upon. Some students were bothered by the limits of the bag sizes, both maximum and minimum. In cooperative mode, a screen lock could occur if two players tried to go in opposite directions. Although we saw this as a method to increase team coordination, it was a source of frustration to 3 particular groups. One group disliked

the layout of the keys, stating it was too crowded. Another student complained of a tired left hand, as it was the student's non-dominant hand.

The final set of issues arose from design and art. Although it was widely enjoyed and praised, there were some points that students did not universally like. The maps were noted as too sparse in one session, both in terms of objects on screen and background design. One student disliked the introduction as they wished to simply get right into the game and did not want to wait through screens of text and pictures. Finally, a different session had a student remark that the overall art style could be seen as offensive to native Americans. The overall art direction was tribal, but that student felt that it was portraying unfair stereotypes. We as a team made sure not to attribute the tribal nature to any particular set of people, especially since our avatars were blue and only mostly humanoid.

5.2.3 Contrasting the Modes and Response to Variables

When asked which mode was easier to play, the students gave varied responses. In 11 of 13 sessions, the students found that the single player mode required less coordination and the player was free to play. On the other hand, the cooperative mode could move faster if the players were well coordinated. Although both of these factors were mentioned by most students, 3 of those 11 used the coordination rationale to select the single player mode as easier while the rest found the cooperative mode easier because of its speed when players coordinated correctly. There was also not a strong correlation between which mode players claimed was easier and which mode students preferred to play. The reasoning stayed the same, but some players enjoyed the version they found to be more difficult. Ease of play and enjoyment of the game did not connect in the minds of our players. However, every student that chose the single player version as their preferred version also mentioned the screen lock as something he or she disliked about the game in general.

Regarding connections to school, most students initially responded by saying that they saw no connection. After some encouragement, students did admit to using basic math. Examples given were multiplication and division to find the value of Herbo types, addition to know how much a particular catch was worth, and multiplication to figure how many of each Herbo was necessary to catch in order to meet goals. Variables were almost never mentioned, even though almost all the students had experienced variables in school. When asked directly regarding variables most students agreed that they were using them, and some even admitted to not realizing it at the time.

As a team we made some observations that did not arise during the post-game questioning. In general, students did not play with much of a perceivable strategy until they began to fail. Most often, this happened on level 4-2, a level requiring students to get exactly 72 points with Herbos worth 8 or 10 points. Students seemed to hit this wall and begin reflecting at the same point regardless of which mode they played. Also, once the student had experienced this turning point in one mode, they were prepared for it in the other mode.

5.2.4 Conversation

We hoped that students would learn through conversation in the multiplayer version. However, conversation did not appear to spur learning any faster in multiplayer than it occurred in single player. One particularly talkative pair did have a lot of conversation, but it was mostly unrelated. The discussion covered wide ranges of topics from their opinions about the game's mechanics to how they played other games with friends. They did not really talk about the goals and strategies until level 4-2.

Other students that played with a student they knew in general did not talk as much as the one pair that had much conversation. Despite that, the conversation that did begin to unearth did tend toward game strategies. One pair said 'This one's 8, 4

is 32.’ This indicated that they were thinking about getting the correct point values. That statement was not made until level 4-2, so the recognition didn’t occur until the barrier of failure had been reached. A different pair of students began talking a bit sooner, in level 3-1. They made the realization that point values did not always change. They pointed out that sometimes point values could be the ‘same thing for each level.’ They also used that fact to develop their strategy for future rounds.

Students that did not test with a friend and instead played cooperatively with a team member did not initiate conversation very much at all. Our team member prodded the students toward the questions that we had wanted them to explore. At least once per session she would ask the student, ‘So what’s our strategy for this level?’ She played very submissively, letting the students drive the progression of the game. Another guiding question given was ‘Is there any difference between the Herbos?’ Students sometimes realized that they were worth different point values, other times they responded with cosmetic differences such as size and color. Even when students mentioned the difference in point values, they did not always seem to formulate a plan using that knowledge.

5.2.5 Strategies

One of our gauges for learning was to ask about the students’ strategies. From a description of a player’s strategy we could try to determine how much understanding of solving for variables was taking place. In 3 of our play sessions, students described a strategy of always catching the higher point Herbos, then catching whatever else was necessary to meet the goal. Although this approach could win many of the rounds, it required some refinement for tougher rounds. Specifically in level 5-2, a player didn’t want to catch any high point Herbos. The goal of level 5-2 was to catch 8 Herbos without exceeding 55 points. Given situations such as that, the ‘whatever else was necessary’ would have to be more clearly defined to work as an overall strategy for

Herbo Hunt.

In order to catch the high point Herbos, the groups must have figured out the point values for the rounds. Although the three that sought out the high point Herbos did not state that as part of their strategy, 6 of the other groups specifically stated this as part of their approach. Some did not elaborate further on how they achieved that goal. One pair of students pointed out that they could catch two of the same Herbo and perform division to find out the individual values. Another student realized the practice round structure of most of the levels, and stated that he would determine the point values during this free round.

There were few other approaches mentioned by the students. 2 groups tried to stay as close to the medimaid hut as possible, in order to turn in more quickly and more often. A different group tried to take separate parts of the map when they played cooperatively, allowing them to catch the Herbos if they were more spread out. One student had a very particular strategy for the rounds with 10-point Herbos: he would catch enough of the Herbos that were not worth 10 points in order to get the ones place of his score to be correct. After getting the correct ones value, he could simply catch all 10-point Herbos until he met his target score.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

As a game, Herbo Hunt is fun, engaging, and served its purpose for this study. Students wanted to continue playing even after their time was up. It gave us two games that differed almost solely in the number of players, as the levels and challenges all stayed the same across the two versions. On its own merits, however, the game does leave some to be desired. The game overall is simply too thin in many aspects that draw in players and entice them to continue. For the purposes of the study, Herbo Hunt proved efficient and effective, but it needs more polish before it can be published onto a website for the public.

After performing the study, we have gained no strong evidence that students were learning more quickly or more deeply when playing in either mode. The learning always seemed to take place on the same level in both games. Due to the similarity in the games, the techniques learned in one version could simply be memorized and recycled. This may have affected the results, but changing the versions considerably would potentially have altered our study in other ways.

Before the study, we believed that the students would talk through the gameplay

and discover the educational concepts through discussion. The discussion that we observed did tend toward understanding of the game, but often did not occur until the failure point had been reached. This is part of what may have contributed to the similarity of results between the two games. Even when prompted by our team members to think about point values and strategies, we did not observe the early strategizing that we had hoped would happen with a cooperative multiplayer mode.

Despite the lack of evidence from this study, we still see potential in cooperative gaming as an educational medium. Students have always been forming study groups and working with teams in order to get a better grasp on concepts. A game that required more explicit coordination between teammates may produce stronger results. One pair of students stood out during the study because they discussed everything about the game. They even seemed to develop a bit of an understanding before failing levels. However, since it was only the experience with a single trial, we cannot conclude this to be fact without more similar results.

6.2 Future Work

There are many opportunities for extension of this study. Areas of further exploration fall into four main categories. Improvements to the game can be made to both alleviate the complaints of students and improve engagement with the game itself. The study can be improved to try to get more accurately controlled and/or true results. The level goals and mechanics could be modified to encourage cooperation or answer other questions. There are other tangential opportunities that are related but not necessarily changes that would aid this particular study.

There are many improvements available for the game Herbo Hunt itself. Some have been mentioned already, but we gather them all here for analysis. The story seemed unfinished, so bringing in story that connected the different levels and wrapped up

the game's ending could increase engagement on the player's side. Early designs of the game's flow had four different parts of a world, and the player traveling to the different locations to help different groups of people. This aspect of the game would keep the experience fresh. Depth of the game was also a concern. Suggestions to improve on this aspect are to add more to the objects that can appear on the map such as obstacles or traps, and adding an in-game store or upgrade system. Online multiplayer was also suggested, both to deal with cramped quarters on a keyboard and to allow players to play with friends that didn't happen to be in the same room.

As already discussed, procedural map generation was a feature that was removed as development proceeded. Although time constraints did not allow this feature to be realized fully, it could still provide depth and replay value to the final product. The random level generation code that worked during the last iteration before the removal is still in the code base. The level generation parameters have not changed much since that point, so it would not take much work to restore this feature.

Despite the fact that restoring the feature may not be too difficult, one of the reasons that this facet of the game was eliminated was the fact that purely random generation is not preferable. Purely random generation could result in, for example, a game that has a very high point target and an impossibly short time in which to do it. The parameters would need to be grouped in some way. Rather than randomizing every parameter independently, the use of decision trees would also make the procedural content generation more beneficial. The game could randomly choose a goal, and then based on that goal it would randomly select the target number from that goal from a subset of all the possible targets. Using this approach, procedural generation could be restored for a future version of the game.

The study could be modified to attain results of a different nature. Changing the evaluation metric, possibly to something like an aptitude test, could give more objective data than that provided by an informal questionnaire and noted observations.

We observed that students communicated very minimally, and that may have affected the differentiation in learning. The facilitators of the study could actively encourage more communication to see if that factor produces a stronger effect on concept understanding. Finally, having different goals between the two versions would prevent memorization of the solutions, and might provide different results when switching versions.

To further explore differences between single player and multiplayer, we could try to create level goals that tried to encourage more cooperation between players. If different players couldn't catch some of the onscreen Herbos while their teammates could, that may bring out more conversation. If the players had to share a bag rather than have two separate bags, they may discuss what to catch beforehand. Another option is to have the Herbos that player 1 turns in affect player 2's score as well as player 1's score. We could also add a Herbo that both players had to surround in order to catch.

We could also try to create more level goals that encouraged thinking about variables, as this may have affected our study. An even larger bag size might prompt students to think further ahead. Having more different Herbos on the map at once might also get students to catch more carefully. A suggestion from a STEP lab member that could also help would be to have different turn in huts on the same map that had different point values for the Herbos. We could also provide a more rapid-fire experience by having point goals on individual turn ins rather than a level point goal. This would hopefully cause players to plan on every catch.

There are many other potential options that could change the final product. A study could be performed seeing if the type of coop partner made a difference. Since the partners here were either friends or a single student playing with Louisa, we only had two particular dynamics to analyze. We could use partners that knew each other, liked/disliked each other, players local or networked, and players with varying skill

levels both in math and in playing video games. Another road that could be taken to different results would be to try to make the gameplay more engrossing to the player. Possibly by immersing the player with a deeper game, the learning could be even more subtle. Suggestions for this include expanding the game as previously described, and experimenting with the avatar selection. By providing more character options, we might be able to provide a closer connection between player and avatar.

Appendix A

Herbo Hunt Level Progression

In Herbo Hunt, the game is separated into worlds. These worlds are further divided into levels. Point values do not change within the same world. The first level of a world is often a practice run that is impossible to fail. This allows the player to determine the values before undertaking a real challenge. Once the bag increases size, it stays that way for all of the remaining worlds. Levels with opponents had a time limit, so players had to constantly compete with the onscreen opponent to achieve the goal.

Table A.1: Levels

World 1 - Introduction			
Level	Goal	Values	# Opponents
1-1	Score more than 15 points	10	0
1-2	Score exactly 50 points	10	0
World 2 - Adds a Second Herbo			
Level	Goal	Values	# Opponents
2-1	Score more than 45 points	10, 5	0
2-2	Score exactly 100 points	10, 5	0
World 3 - Point Values Change			
Level	Goal	Values	# Opponents
3-1	Score more than 55 points	8, 10	0
3-2	Score exactly 72 points	8, 10	0
World 4 - Bag Size Increases to 2			
Level	Goal	Values	# Opponents
4-1	Score more than 55 points	8, 10	0
4-2	Score exactly 72 points	8, 10	0
World 5 - Catching Efficiency			
Level	Goal	Values	# Opponents
5-1	Score ≥ 60 points, catch ≤ 8 Herbos	10, 5	0
5-2	Score ≤ 55 points, catch ≥ 8 Herbos	10, 5	0
World 6 - Negative Point Values			
Level	Goal	Values	# Opponents
6-1	Score more than 100 points	10, 5, -5	0
6-2	Lose 100 points	10, 5, -5	0
World 7 - Play an Opponent			
Level	Goal	Values	# Opponents
7-1	Score the Most Points	2, 5, 10	1
7-2	Get the most Herbos and fewest points	2, 5, 10	1

Appendix B

Demonstration and Playtest Feedback

B.1 Demonstration Notes

- There was a lot of support behind the baskets idea. Rather than a particular scoreboard update rate, instead have the player turn in what they catch in an individual trip. Player can get a detailed description, or no real feedback other than points obtained.
- The suggested time for the most reflection would be between rounds, especially on failed attempts. The player could even be given tips on how to succeed the next time s/he played.
- There were supporters of the fast-twitch aspect, while others pushed for a slower approach. One suggestion was a point-and-click turn based game. Another suggestion would be to have everybody take one step whenever the player stepped. Keeping the player onscreen was preferred if this were to ever make the jump to STEMMO.
- Monster differentiation caused some discussion. Differing the color was recommended as the easiest way to tell the monsters apart. Another level of differentiation could be achieved by having heads and bodies that can be mixed and matched, however this would be harder for players to search for.
- Progression came up in a few different ways:
 - Start with no opponents, bring them in later. This progression could be taken every series, with a test round of no opponents followed by competitive rounds. Alternatively, it could be an overall progression from personal goals to competitive challenges.
 - With the basket idea, the player could be given different basket receivers. This would allow for different point values for the same monster. It could also give the ability to have more or less feedback on basket reception.

- Keeping point values the same for a number of rounds was suggested, this makes a better case for it as the series approach.
- Although maybe not constantly onscreen, it is very important to give the player knowledge of just what monsters s/he is catching. Suggestions were to have it on basket turn-in, to have it at the end of rounds, or to have it constantly onscreen.
- There was a suggestion of having some kind of tool to perform the algebra, like using a Punnett square in biology. No real tool was recommended, but it was a suggested approach.
- Putting a Mastermind-like recap of baskets turned in was suggested, so the player could quickly look over what combinations yielded what point values.
- Change the scores by using a Stock Market.
- Give the monsters different sizes, and have a basket capacity, forcing the player to choose.
- Had a question regarding ensuring learning rather than allowing luck and perseverance of bad (or nonexistent) strategies. Looking at other games could be helpful in this area. A special warning on least points, most monsters that a player could catch 1 and take his/her chances that it was the lowest point monster. Although the system is not purely an average, the player could still do fairly well catching only one monster.

B.2 4/9/12 Playtest Feedback

B.2.1 Playability

- Player reads instruction. When he sees downward triangular arrow in the intro screen, it's not obvious to press "Enter".
- Sometimes player makes mistake and knows that he can't win the round, but he still needs to go through the whole round. Is there a way to redo a move or terminate a round?
- Player keeps forgetting to have the bag full before turning in after bag size has increased.
- Players say their favorite part is the increase in bag size. Maybe the first two rounds can be shortened a bit, so that the player can get to the fun part faster?
- Got started playing pretty easily
- Had trouble turning in when he wasn't walking into the box, etc.

- Round 2-2 took longer but was fine - he said his strategy was to just get the 10s so he didn't have to worry about the others
- Level 3-2 had only greens one time, then only blues. bug?
- 4-2: went over because she wasn't watching closely, this one is tricky
- Figured out the turnin easily enough
- Tough round to get exactly 72 with 2 monsters at a time
- Monsters would disappear without catch
- Need to move toward monster on catch

B.2.2 Controls

- Player naturally goes for the mouse rather than the keyboard in the beginning.
- Player keeps forgetting to use keyboard and always goes back to clicking on the cursor.
- Was fine with giving up the mouse to play the game.
- Forgot to use L to catch and was confused at first.
- Hit shift by accident a lot and paused the game.
- Tried the mouse on turnin.

B.2.3 Information

- It may be helpful to give reason for losing.
- When player sees "Level 2" screen, there is a sense of accomplishment.
- Player is not sure what monsters are caught. Player needs to remember the amount mentally.
- Player doesn't know that after bag size is increased, it must contain exactly two monsters before turning in.
- Didn't know how to create a new user.
- At first he said the goal was to get as many points as possible, then he realized he had to get the exact score.
- When the point values changed, he didn't realize for a while that they had.
- Understood the point values and goals, no big problems.

- Level 2-2: went over 100 and had no idea why she'd lost (didn't understand what the goal was or that she should watch the points as she went) but figured it out when prompted to think it over.
- Level 3-1: at first didn't notice the point changes but then was able to figure it out.
- Number of animals vs Number of turnins unclear.
- Didn't know L could remove summary.
- Didn't understand break score Vs exact score.

B.2.4 Design

- Player asks for "how many lives" are left.
- Players do not mind walking through the map. It gives time for them to remember and think through things.
- Players express that the setting should change after each level.
- Players see points as money, and suggest to increase the amount of values/points as level increases.
- Players suggest to make monsters harder to catch at higher levels, and also to have bigger monsters (e.g. dinosaurs, dragons) at higher levels.
- Players said that the game is like an addition game.
- Liked it when he discovered he could walk beyond the edge of the screen.
- Wanted more different kinds of goals/rounds.
- When he reloaded the page, he expected to start at the beginning of the same round but he had to start the whole level over again.
- When bag size went up, it was exciting. At first he thought you could still turn in 1 monster at a time but quickly figured it out.
- Suggestions: different maps, harder math, lives that you lose due to traps, etc. on the level, when all lives are gone you restart that whole level.
- Suggestion: swap points for money to upgrade avatar.
- L double-tap eliminated summary.

B.2.5 Style

- Player comments “It’s like Pac-man.”
- Players describe the avatar as “funny-looking”, “cartoony”, and imagine that the avatar hunts.
- It is obvious the guy avatar is a guy.
- Players like monster, but one says he wouldn’t want to eat it. Both players imagine the monster to live in a grassland.
- Players would like to catch the monster.
- Players suggest to have avatars look less mean (e.g. put a smile on), and comments that the monster looks like a little kid.

B.3 4/28/12 Playtest Feedback

B.3.1 Observer 1 Notes

- When she started playing, one girl said: “Is this one of those addicting games?”
- One girl said she didn’t like mathy games and this one didn’t seem very mathy.
- One asked how it was math-related - they got a better understanding of it later but it was nice that it didn’t feel like a math game right off the bat.
- IMPORTANT: a bunch of pairs didn’t realize that the two-player version meant they were playing as a team. they kept asking which of them had won.
- Said the game needed more of a point to it, meaning things like storyline, obstacles, bad guys, bonuses, etc.
- Player 2 using G to catch was awkward because it’s right in the middle of the keyboard and they couldn’t really both get in there.
- In multiplayer version, the score says P1 score - should change to Team or something.
- A couple pairs played single player as multiplayer, with one girl doing the arrows and the other girl catching with L - interesting way to collaborate.
- One girl explained that you had to solve each level by using the values of the animals - some got it very clearly.
- One girl said “we have to see if they’re all the same though” meaning the animals’ point values - other girls had trouble thinking of that idea

- One team had an epiphany when they said “maybe we need to get the fat ones because the fat ones are 10!”
- Avatars should look more different so you can tell them apart more easily.
- Wanted to earn something for winning, like feathers in your headdress, and then bad guys could take the feathers away.
- Wanted to see the ones you caught later on, maybe after you win the round.
- 1 pair wanted a split screen multiplayer so you could walk around freely.
- They liked the animals caught restrictions because it made it more challenging.
- The pop ups definitely slowed them down and they felt it.
- Wanted powerups that let you get a x2 bonus, etc.

B.3.2 Observer 2 Notes

- Avatar walks funny.
- Needs a story to go along with the game.
- Make the map into a maze to create obstacles.
- Players like the new font.
- Create a monster/bad guy, that if you catch it, you lose all your points.
- The background should change color.
- The background should have a forest and obstacles.
- Should have obstacles when chasing monsters.
- They like the jumping of the monster - makes it more challenging.
- Should have something new for avatar at high levels, e.g. adorning new feather, accessories, head dresses.
- Cute name for avatar: “avadagoo.”
- Names for monsters: “globbos”, “cherubs.”
- Could have two sacs to turn in and evenly distribute the monsters.
- Needs a summary of all the monsters you caught at the end of each round.
- Instruction is clear in the beginning.
- Have expression on monsters.

- Name for game: “Cherokee Hunt”, “Cupcake and Flowers.”
- Want to customize avatar.
- Points as money to buy things, or power up.
- Suggest to have time limit.
- Certain monsters will only appear for a set amount of time.
- Catch monsters in a certain order.

B.3.3 Discussion Responses

- Players thought animation was cool.
- Players liked the need to solve, in particular when they need to score a certain number of points with fewer than some number of monsters. It’s more interesting/challenging.
- Players wanted more background colors.
- Players felt score pop up slowed them down.
- Players said score pop up gives useful info for the first time but having it pops up every time when turning in monster is annoying.
- Players see how multiplication/division, and addition are used in the game.
- It was not very obvious at first that it’s an education game.
- Game is not teaching something new/ didn’t learn anything.
- Players suggested competitive multiplayer.
- 2 players setting is too crowded - keep bumping into each other, restrict motion on screen, keep running into each other and getting stuck.
- When one player turns in, the score screen pops up, and that interrupts the other player.

Bibliography

- [1] Compton, K. and Mateas, M. Procedural level design for platform games, in Proc. 2nd Artif. Intell. Interact. Digital Entertain. Conf., Marina del Rey, CA, 2006, pp. 109111.
- [2] Mitchell, A. and Savill-Smith, C. (2005). The use of computer and video games for learning: A review of the literature. London. Learning and Skills Development Agency.
- [3] Rosas, R., Nussbaum, M., Cumsile, P., Marianov, V., Correa, M., Flores, P., Grau, V., Lagos, F., Lpez, X., Lpez, V., Rodriguez, P., Salinas, M. (2003). Beyond Nintendo: design and assessment of educational video games for first and second grade students. *Computers and Education*, 40, 7194.
- [4] Steinkuehler, C. A. (2004b). Learning in massively multiplayer online games. In Y. Kafai, W. A. Sandoval, N. Enyedy, A. S. Nixon & F. Herrera (Eds.), *Proceedings of the Sixth International Conference of the Learning Sciences* (pp. 521528). Mahwah, NJ: Erlbaum.
- [5] "STEP History and Mission." MIT STEP. STEP, n.d. Web. 26 Nov. 2012. <<http://education.mit.edu/classes/step-history-and-mission> >.
- [6] Virvou, M., Katsionis, G., & Manos, K. (2005). Combining software games with education: Evaluation of its educational effectiveness. *Educational Technology & Society*, 8(2), 54-65.